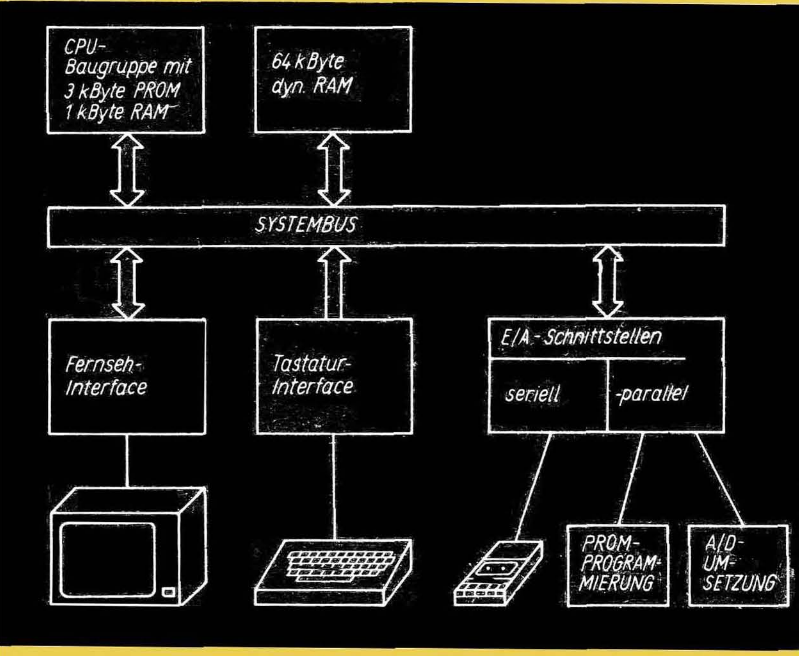
amateurreihe

electronica



Bernd Hübler
Klaus-Peter Evert

227/228

Ausbaufähiger Mikrocomputer mit dem U 880

electronica · Band 227/228

BERND HÜBLER · KLAUS-PETER EVERT †

Ausbaufähiger Mikrocomputer mit dem U 880

MILITÄRVERLAG
DER DEUTSCHEN DEMOKRATISCHEN
REPUBLIK

Bernd Hübler/Klaus-Peter Evert:
Ausbaufähiger Mikrocomputer mit dem U 880.
– 1. Auflage – Berlin: Militärverlag der DDR, 1985 –
(electronica 227/228)

1. Auflage, 1985

Lizenz-Nr. 5
Printed in the German Democratic Republic
Gesamtherstellung: Druckerei Märkische Volksstimme Potsdam
Lektor: Rainer Erlekampf
Zeichnungen: Angelika Ulsamer
Typografie: Martina Schwarz
Redaktionsschluß: 12. November 1984
LSV 3539
Bestellnummer: 746 728 1
00380

Inhaltsverzeichnis

1.	**Vorwort**	5
2.	**Systembeschreibung**	7
3.	**CPU-Baugruppe**	10
3.1.	Taktversorgung, Buspufferung	10
3.2.	Speicher	14
3.3.	Zentrale Funktionseinheiten	15
3.3.1.	Speicherselektierung	15
3.3.2.	I/0-Portselektierung	17
3.3.3.	Startlogik	18
3.4.	Aufbau und Inbetriebnahme	22
4.	**Speichererweiterung**	25
4.1.	ROM-Speicher	26
4.2.	RAM-Speicher	30
4.2.1.	Statischer RAM-Speicher	30
4.2.2.	Dynamischer RAM-Speicher	32
4.3.	Aufbau und Inbetriebnahme	38
5.	**Fernsehinterface**	42
5.1.	Zeichendarstellung auf dem Bildschirm	42
5.2.	Schaltungsbeschreibung	46
5.2.1.	Bildwiederholspeicher, Adreßmultiplexer	48
5.2.2.	Synchron- und Austastsignalerzeugung, Adressenbereitstellung	52
5.2.3.	Videosignalerzeugung	55
5.2.4.	BAS-Signalerzeugung	61
5.3.	Erweiterungsmöglichkeiten	61
5.3.1.	Zeichengenerator mit Graphiksymbolen	62
5.3.2.	RAM-Speicher als Zeichengenerator	62
5.3.3.	Vollgraphik	62
5.4.	Aufbau und Inbetriebnahme	63
6.	**Alphanumerische Tastatur**	65

6.1. Schaltungsbeschreibung 65
6.2. Aufbau und Inbetriebnahme 70

7. Ein-/und Ausgabebaugruppen 74
7.1. Universelle I/O-Karte 74
7.2. EPROM-Programmiergerät 79
7.2.1. Programmiervorschriften 79
7.2.2. Die Schaltung 81
7.2.3. Aufbau und Codierung 83
7.2.4. Ein Beispielprogramm 85
7.3. Analogschnittstellen 87
7.3.1. Digital/Analog-Wandler 87
7.3.2. Analog/Digital-Wandler 91

8. Kassetteninterface 95
8.1. Aufzeichnungsverfahren 95
8.2. Realisierung 97

9. Monitorprogramm 104
9.1. Beschreibung der Monitorkommandos 104
9.2. Schnittstellen und Erweiterbarkeit 110
9.3. Das Programm 113

10. Hardwareaufbau und Inbetriebnahme 177
10.1. Stromversorgung 177
10.2. Systemaufbau 179

11. Anhang 184

12. Literaturverzeichnis 188

1. Vorwort

Die Möglichkeiten und Erfordernisse der modernen Mikroelektronik zeigen sich im Bereich der Mikrocomputertechnik besonders deutlich.

Der Anwendungsbereich des Mikrocomputers reicht von der Steuerung und Regelung industrieller Prozesse – genannt sei hier nur das Schlagwort Industrieroboter – über den wissenschaftlich-technischen und den Bürocomputer bis hin zum Heimcomputer. Immer stärker dringt der Mikrocomputer auf Grund seiner ständig wachsenden Leistungsfähigkeit in die Bereiche der traditionellen EDV ein.

Für ein modernes Industrieland ist es unumgänglich, die Möglichkeiten dieser neuen Technik zu nutzen.

Wissen über Computertechnik im allgemeinen und Mikrocomputertechnik im speziellen sowie computerspezifische Denkweisen dürfen deshalb nicht einem kleinen Kreis von Spezialisten vorbehalten bleiben. Diese Erkenntnis hat international zu einer bemerkenswerten Verbreitung und Popularisierung des Heimcomputers bzw. des Personalcomputers besonders bei jugendlichen Interessenten geführt.

Das vorliegende Heft wendet sich deshalb an den elektronisch versierten Amateur, aber auch an den Elektroniker, der noch keine Gelegenheit hatte, auf dem Gebiet der Mikrocomputertechnik praktische Erfahrungen zu sammeln. Grundkenntnisse der Digitaltechnik sowie der Mikroprozessortechnik, wie sie beispielsweise [1] vermittelt, werden vorausgesetzt.

Das gegenwärtig in der DDR verfügbare Schaltkreisangebot gestattet dem potentiellen Interessenten den Aufbau eines leistungsfähigen Mikrocomputers. Nachbaufähige Schaltungen für die Realisierung der dazu benötigten Hardware sowie erprobte Betriebssoftware sollen den Weg zum fertigen Gerät erleichtern.

Der Aufbau des Systems ist mit einiger Mühe verbunden und erfordert ein gewisses Durchstehvermögen.

Beim eigenen Realisieren begreift man jedoch die relativ komplizierte Hardware am besten. Gerade für den Bereich des Mikro-

computers sind gründliche Hardwarekenntnisse von besonderem Wert.
Die Autoren hoffen, daß einem möglichst breiten interessierten Personenkreis der Einstieg in die zukunftsträchtige Problematik der Mikrocomputertechnik ermöglicht wird.

Berlin, im Januar 1984

Bernd Hübler
Klaus-Peter Evert †

2. Systembeschreibung

Das in der Beschreibung vorgestellte Mikrocomputerkonzept basiert auf der Schaltkreisfamilie des Mikroprozessors *U880*. Diese leistungsfähige 8-Bit-CPU einschließlich der dazugehörigen Peripherieschaltkreise sowie der bereits verfügbaren Speicher erlaubt bei vertretbarem Aufwand den Aufbau eines universell einsetzbaren Mikrocomputers.
Bild 2.1 zeigt den Übersichtsschaltplan der Hardware des Systems. Danach besteht das voll ausgebaute System aus folgenden Komponenten:

- CPU-Baugruppe (realisiert als Einplatinenrechner) mit 3-kByte-PROM, 1-kByte-RAM
- 64-kByte-RAM-Speicher
- Bildschirm mit 24 Zeichenzeilen zu je 64 Zeichenpositionen
- alphanumerische Tastatur (maximal 64 Tasten)
- Kassette als externes Speichermedium
- parallele und serielle Ein-/Ausgabeschnittstelle
- PROM-Programmiergerät

Als Bildschirm eignet sich ein handelsüblicher Schwarzweißfernsehempfänger (möglichst Koffergerät), und zur Bandaufzeichnung nutzt man einen Kassettenrecorder.
Die Hardware kann schrittweise, beginnend mit Aufbau und Inbetriebnahme des Einplatinenrechners bis hin zum kompletten System, ausgebaut werden.
Das Systemkonzept des beschriebenen Mikrocomputers soll einen möglichst universellen Anwendungsbereich sichern. Dementsprechend kann man ein breites Softwarespektrum einsetzen. Es reicht vom Maschinensprachniveau bis zu Assemblerprogrammen und zu höheren Programmiersprachen (z. B. BASIC). Aus diesem Grund steht der RAM-Speicher ab Adresse 0 zur Verfügung. Ein möglichst weitgehender Ausbau des RAM-Bereichs erweist sich unter den genannten Aspekten als günstig.
Der Computer soll aber auch als Entwicklungssystem genutzt werden. Für diesen Anwendungsfall wird ein Gerät vorgestellt, das die Programmierung aller wichtigen, derzeit bekannten PROM-Typen ($\geqq$ 1 kByte) gestattet.

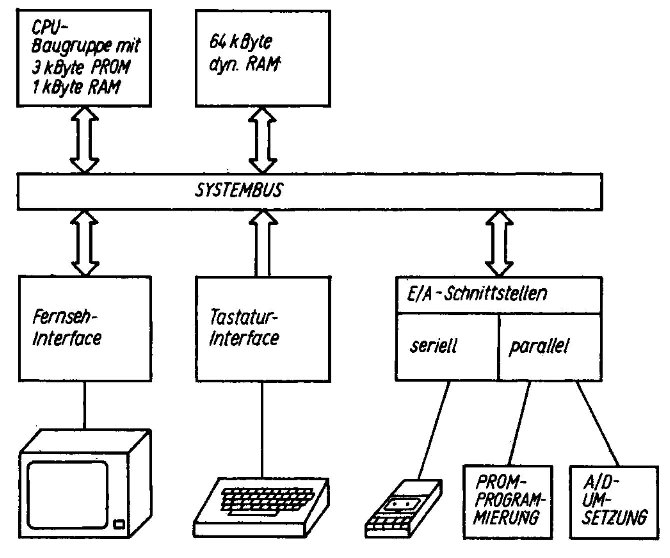


Bild 2.1 Hardwareübersicht

Voraussetzung für die Arbeit mit dem Computer ist das Vorhandensein eines (im allgemeinen) systemresidenten Betriebssystems. Als Minimalforderung muß ein Urlader auf dem System abgespeichert sein, der das Laden eines Betriebssystems von einem externen Speichermedium ermöglicht. Für den vorgestellten Mikrocomputer wurde ein den Belangen des Systems angepaßtes Monitorprogramm entwickelt. Mit Hilfe des Monitors wird der Datenverkehr zwischen CPU und Peripherie abgewickelt und das Arbeiten auf Maschinensprachniveau ermöglicht. Es lassen sich also Programme sowie Datenein- und -ausgaben testen. Darüber hinaus verfügt das Monitorprogramm über Schnittstellen, die es gestatten, Teile desselben in Anwenderprogrammen mitzunutzen. Beispielsweise können die auch beim Betrieb mit höheren Programmiersprachen benötigten Treiberroutinen (Bildschirm, Tastatur, Kassettenaufzeichnung) so in einfacher Weise angesprochen werden.

Unter Berücksichtigung des Systemkonzepts sollte der Monitor das einzige systemresidente, d. h. auf PROM abgespeicherte, Pro-

gramm sein. Alle weitere Software kann vom externen Speichermedium (Kassette) bei Bedarf in den Arbeitsspeicher des Mikrocomputers geladen werden. Abhängig von den Möglichkeiten (z. B. hinsichtlich des RAM-Speicherausbaus) lassen sich jedoch auch individuelle Lösungen realisieren. Beispielsweise kann ein geeigneter BASIC-Interpreter auch auf PROM abgespeichert werden. Die schaltungsmäßigen Lösungen für den in diesem Fall zusätzlichen Bedarf an PROM-Speicherkapazität sind im vorgestellten Hardwareumfang enthalten.
Die frei verfügbaren E/A-Schnittstellen gestatten das Koppeln des Mikrocomputers mit externen Geräten. Für die in diesem Zusammenhang häufig benötigten A/D-Wandler bzw. D/A-Wandler werden Lösungsvorschläge unterbreitet.
Im voll ausgebauten Zustand steht damit ein leistungsfähiger Mikrocomputer zur Verfügung, der darüber hinaus hard- und softwaremäßig für individuelle Erweiterungen vorbereitet ist.

3. CPU-Baugruppe

Die CPU-Baugruppe stellt, wie bereits der Name sagt (*Central-Processing-Unit* – zentrale Verarbeitungseinheit), das Herzstüsck des Mikrocomputersystems dar. Basierend auf dem Mikroprozessor *U 880* , bestimmt sie in wesentlichem Umfang die Leistungsfähigkeit des Systems.
Als Voraussetzung zur Erfüllung ihrer Hauptfunktion hat die CPU-Baugruppe die Aufgabe der Taktversorgung des Mikroprozessors sowie der Verteilung der Systeminformationen (Daten, Adressen, Steuersignale) im Computer. Ihre spezielle Konfiguration ist vorrangig vom gewünschten Einsatzzweck (z. B. Heimcomputer, komfortabler Bürocomputer, Steuercomputer usw.) abhängig und damit auch vom Umfang und Niveau der Software, die man beabsichtigt, auf dem Computer lauffähig zu machen (fachmännisch ausgedrückt: zu implementieren). Letztlich ist die CPU-Platine auch eine Widerspiegelung der Systemphilosophie des Entwicklers.
Bild 3.1 zeigt die für diese Broschüre gewählte Variante.

3.1 Taktversorgung, Buspufferung

Als Taktgenerator wird ein quarzstabilisierter TTL-Generator verwendet. Er schwingt mit 10 MHz und liefert der CPU, nachdem diese Frequenz 1 : 4 untersetzt und durch 2 Leistungsgatter *D 240* gepuffert wurde, den Standardtakt von 2,5 MHz. Außerdem wird der Takt am Systembus bereitgestellt.
Wer über einen geeigneten Schwingquarz verfügt, hat die Möglichkeit, mit der angegebenen Schaltung oder entsprechend anders eine Taktfrequenz von genau 2,4576 MHz zu erzeugen. Damit lassen sich standardisierte Datenübertragungsraten (Baudraten) für die entsprechende Peripherie (Kassettenaufzeichnung, Datenfernübertragung usw.) realisieren.
Die an den $\overline{\text{RESET}}$-Eingang des Mikroprozessors angeschlossene Kombination aus *RC*-Glied und den Gattern D3.1 und D3.2 bewirkt das Einschalt-RESET (Power-On-RESET) und damit den

Urstart des Systems. Diese $\overline{RESET}$-Schaltung wird häufig auch mit einfachen NAND-Gattern bzw. ohne Gatter ausgeführt. Die Autoren empfehlen die Verwendung des Triggerschaltkreises *MH 7413*. Auch das $\overline{RESET}$-Signal steht am Systembus zur Verfügung. Zum Realisieren einer Reset-Taste kann man eine Schaltungslösung verwenden, die einen Datenverlust bei eventuell eingesetzten dynamischen RAM (siehe Abschnitt 4.2.2.) verhindert. Eine mögliche Variante ist [2] zu entnehmen. Die restlichen CPU-Eingänge $\overline{INT}$, $\overline{NMI}$, $\overline{BUSRQ}$, $\overline{WAIT}$ – ihre Bedeutung wird als bekannt vorausgesetzt und ist z. B. in [1], [2] nachzulesen – werden, da sie L-aktiv sind, für den Fall des nicht angesteuerten Zustandes mit den entsprechenden Widerständen auf ein sicheres H-Potential gebracht.

Die Daten- und Adreßausgänge der CPU *U 880* sind als Tri-State-Ausgänge ausgeführt. Sie sind ebenso wie die L-aktiven Steuerausgänge – von denen nur die Signale $\overline{RD}$, $\overline{WR}$, $\overline{MREQ}$, $\overline{IORQ}$ über Tri-State-Ausgänge verfügen – in der Lage, eine TTL-Lasteinheit zu treiben. Für das Vergrößern der Busbelastbarkeit aus der Sicht der CPU sowie zur Leistungstreibung, d. h. zum Unterdrücken des Einflusses von Störimpulsen und parasitären Kapazitäten, muß man die genannten Signale vor dem Bereitstellen auf den geeigneten Treiberschaltkreis puffern.

Der Datenverkehr zwischen CPU und Peripherie ist grundsätzlich bidirektional, d. h., er erfolgt in beiden Richtungen.

Dementsprechend werden die Datenausgänge D0 ... D7 mit 2 bidirektionalen Treiberschaltkreisen des Typs *8216* (D11, D12) gepuffert. Dazu ist es notwendig, gleichwertige Datenein- und -ausgänge (DI und DO) des *8216* parallelzuschalten. Für das Durchschalten in der jeweils gewünschten Richtung, d. h. für das Bereitstellen des Datenrichtungssignals $\overline{DIEN}$ am Schaltkreis *8216*, ist die Datenrichtungslogik, bestehend aus dem Gatter D4.3, zuständig. Wenn das Lesesignal $\overline{RD}$ aktiv ist, ist also der Datenfluß in der Richtung Peripherie →CPU möglich. Für den Fall $\overline{RD}$ = H sind die Verhältnisse umgekehrt.

Das $\overline{M1}$-Signal muß bei der Datenrichtungsumschaltung berücksichtigt werden, damit der Interruptmode 2 (Vektorinterrupt) abgesichert ist. Dabei muß die CPU einen Interruptvektor von der Peripherie lesen. In diesem Fall ist $\overline{RD}$ nicht aktiv, sondern $\overline{M1}$ und $\overline{IORQ}$.

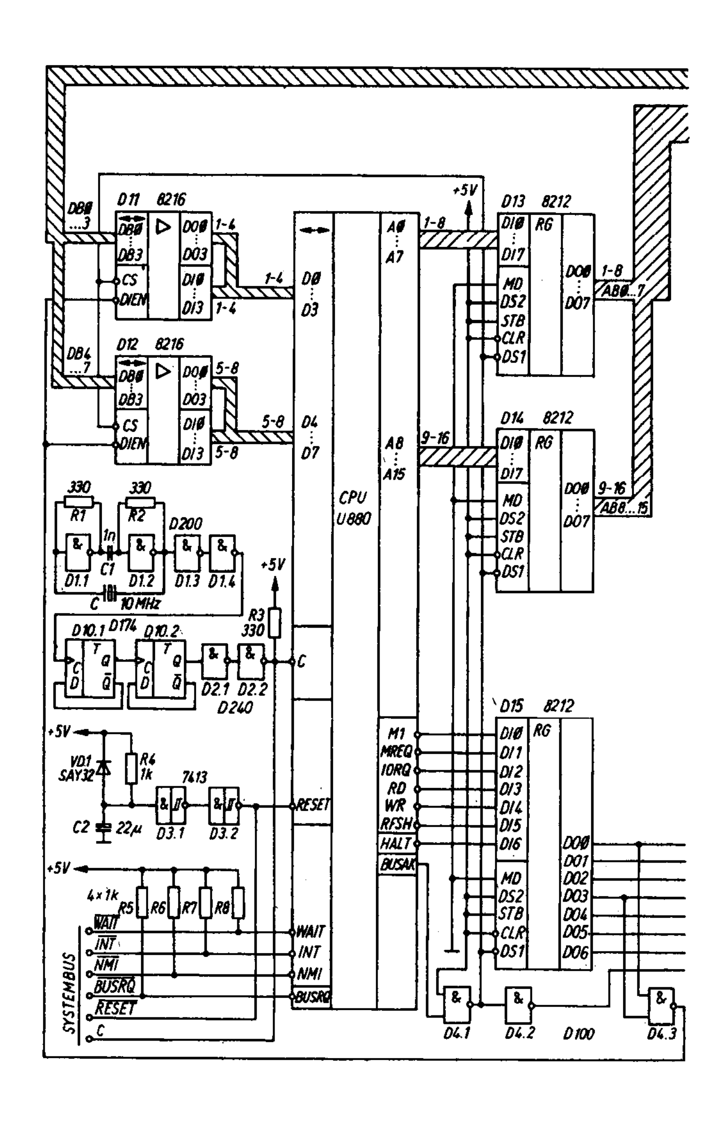
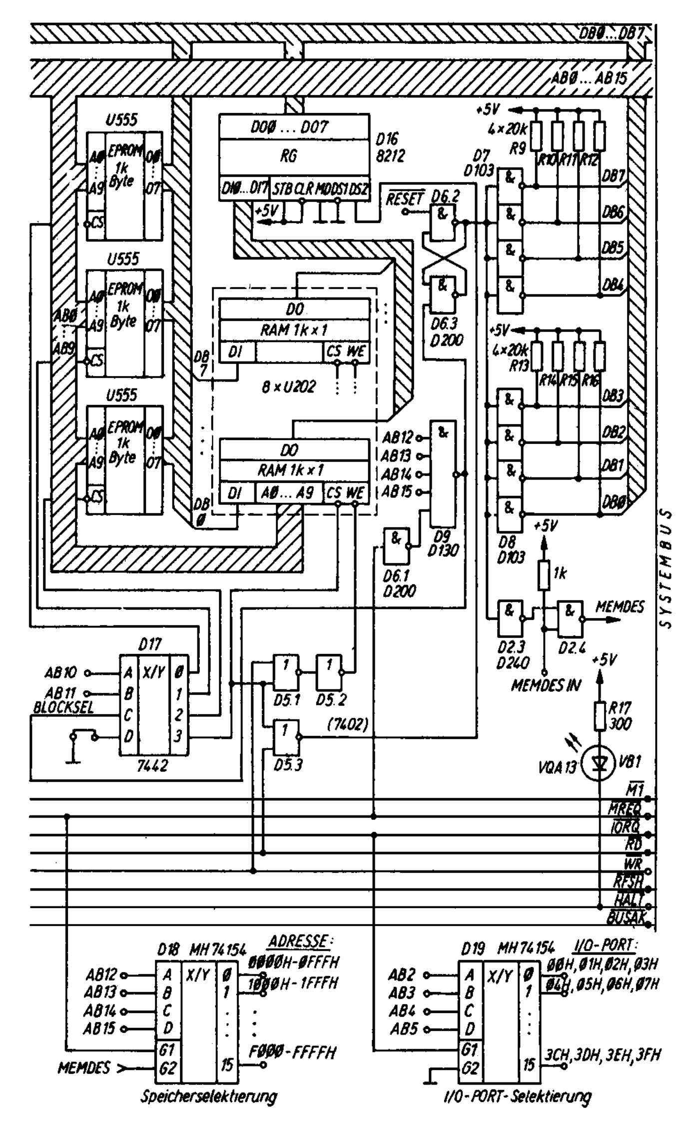


Bild 3.1 CPU-Baugruppe


DB0 ... DB7
AB0 ... AB15
U555
EPROM 1k Byte
D00 ... D07
RG
D16 8212
DI0...DI7 STB CLR MD DS1 DS2
+5V
RESET
D6.2
D7 D103
4×20k
R9 R10 R11 R12
DB7
DB6
DB5
DB4
D6.3 D200
RAM 1k×1
8×U202
DI A0 ... A9 CS WE
AB12 AB13 AB14 AB15
D9 D130
D6.1 D200
R13 R14 R15 R16
DB3 DB2 DB1 DB0
D8 D103
1k
D2.3 D240
D2.4
MEMDES
MEMDES IN
SYSTEMBUS
D17
AB10 AB11 BLOCKSEL
7442
D5.1 D5.2 D5.3
(7402)
R17 300
VQA 13
VB1
M1 MREQ IORQ RD WR RFSH HALT BUSAK
D18 MH 74154
ADRESSE:
0000H-0FFFH
1000H-1FFFH
F000-FFFFH
Speicherselektierung
D19 MH 74154
I/O-PORT:
00H, 01H, 02H, 03H
04H, 05H, 06H, 07H
3CH, 3DH, 3EH, 3FH
I/O-PORT-Selektierung

Als Treiber für die Adreßausgänge A0 ... A15 der CPU setzt man Schaltkreise des Typs *8212* (D13, D14) ein.
Der Schaltkreis *8212* besteht im wesentlichen aus einem 8-Bit-Datenregister und -treiber mit Zusatzlogik sowie Tri-State-Ausgangsstufen. Er ist auf Grund seiner Konzeption (löst auch über ein internes Flip-Flop Interrupt aus) universell zur Bussteuerung und Datenpufferung geeignet.
Im vorliegenden Einsatz ist der *8212* eigentlich unterfordert. Er wurde aus Gründen der guten Verfügbarkeit trotzdem eingesetzt.
Die Steuersignale $\overline{M1}$, $\overline{RD}$, $\overline{WR}$, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{HALT}$ sowie das Refresh-Signal, das man zum „Auffrischen" der dynamischen RAM benötigt, werden über einen weiteren Schaltkreis des Typs *8212* gepuffert.
Mit dem Signal $\overline{BUSAK}$ = L quittiert die CPU die Anmeldung einer DMA-Anforderung am Eingang $\overline{BUSRQ}$. Für diesen Fall sind die Daten- und Adreßausgänge sowie die Steuersignalausgänge $\overline{RD}$, $\overline{WR}$, $\overline{MREQ}$, $\overline{IORQ}$ hochohmig und die weiteren Ausgänge inaktiv. Beim DMA-Betrieb, d. h. beim direkten Speicherzugriff *(Direct Memory Access)* einer peripheren Einheit, ist die CPU bekanntlich unbeteiligt. Deshalb muß sie vom Bus abgeschaltet werden. In der Schaltung gemäß Bild 3.1 wird der aktive Zustand von $\overline{BUSAK}$ am Ausgang von Gatter D 4.1 zum Deselektieren der Bustreiber *8216* und *8212* benutzt. Damit sind alle Daten-, Adreß- und Steuerleitungen (außer $\overline{BUSAK}$) hochohmig vom Bus getrennt.

3.2. Speicher

Bild 3.1 zeigt, daß die vorgestellte CPU-Baugruppe mit 3-kByte-Programmspeicher und 1-kByte-Arbeitsspeicher ausgerüstet ist. Zusammen mit den bereits beschriebenen Schaltungselementen steht damit also ein in sich funktionsfähiger Einplatinencomputer zur Verfügung.
Der Anschluß der verwendeten EPROM-Schaltkreise *U 555 (2708)* sowie RAM-Schaltkreise *U 202 (2102)* an den Daten- und Adreßbus geht ebenfalls aus Bild 3.1 hervor. Man stellt fest, daß eine Zusammenstellung gleichwertiger Datenein- und -ausgänge (DI, DO) beim RAM *U 202* nicht möglich ist. Aus diesem Grund werden die Datenausgänge DO nur für den Fall, daß die CPU ei-

nen Speicherlesezyklus zum selektierten RAM-Block durchführt, über die Tri-State-Ausgänge des Schaltkreises *8212* (D15) auf den Datenbus geschaltet. Andernfalls sind sie hochohmig vom Bus getrennt. Günstig, besonders wenn man größere Speicherblöcke auf der Basis statischer RAM des Typs *U 202* verwendet, ist das Puffern sowohl der Dateneingänge als auch der Datenausgänge. Hierfür bietet sich wieder der bidirektionale Treiberschaltkreis *8216* an. Die Verdrahtung der benötigten Schaltkreise *8216* mit den Ein- bzw. Ausgängen des *U 202* ergibt sich aus den jeweiligen Pin-Belegungen (siehe auch Bild 4.3).
Die Lese/Schreib-Umschaltung am $\overline{\text{WE}}$-Eingang des *U 202* sowie die Bereitstellung des für die gewünschte Funktion des *8212* notwendigen Signals DS2 realisiert man mit geringem Aufwand über die NOR-Gatter D 5.1, D 5.2, D 5.3. Dazu werden die CPU-Signale $\overline{\text{RD}}$, $\overline{\text{WR}}$ verwendet.
Beim Einsatz des *8216* steht als Datenrichtungssignal $\overline{\text{DIEN}}$ das $\overline{\text{RD}}$-Signal zur Verfügung (siehe auch Bild 4.3).
Für die Chipselektierung der 3 EPROM *U 555* sowie des 1-kByte-RAM-Blocks ist eine 1-aus-4-Decodierung notwendig. Sehr einfach und übersichtlich läßt sich dies mit dem 1-aus-10-Decoder *MH 7442* lösen, der das jeweilige Chipselektsignal aus den Adressen AB10, AB11 sowie einem 4-kByte-Blockselektsignal decodiert. Aber auch jeder andere 1-aus-n-Decoder ($n \geqq 4$) ist geeignet (z. B. 1-aus-8-Decoder *8205*).
Das CS-Signal für die eventuell eingesetzten bidirektionalen Treiber *8216* ist mit dem für den 1-kByte-RAM-Block identisch.

3.3. Zentrale Funktionseinheiten

Die 3 auf der CPU-Platine untergebrachten Funktionseinheiten haben in engerem Sinne mit der Funktion der CPU-Baugruppe nichts zu tun, sie sind aber für das Systemkonzept von Bedeutung.

3.3.1. Speicherselektierung

Die Speicher werden bei üblichen Mikrocomputersystemen (z. B. dem *K 1520* vom VEB Kombinat *Robotron*) häufig dezentral se-

lektiert, d. h., mit entsprechenden Decodierschaltungen erzeugt sich jede Speicherplatine ihren Adreßbereich im Adreßraum des Computers (64 kByte beim *U 880*) gewissermaßen selbst. Dieser Adreßbereich ist meistens vom Anwender über Drahtbrücken oder DIL-Schalter auswählbar.

Im Gegensatz dazu wurde für das vorgestellte System eine Schaltungsvariante zur zentralen Speicherselektierung entwickelt. Als elegante Lösung bietet sich der 1-aus-16-Decoderschaltkreis *MH 74154* an, der es erlaubt, aus den Adressen AB12 ... AB15 16 4-kByte-Blockselektsignale zu decodieren. Damit ist man in der Lage, den 64-kByte-Adreßbereich der CPU *U 880* in 16 4-kByte-Speicherblöcke aufzuteilen. Tabelle 3.1. verdeutlicht dies.

Für die praktische Auswahl eines gewünschten Speicherbereichs ist es lediglich notwendig, den jeweiligen Ausgang des *MH 74154* mit dem Blockselekteingang der entsprechenden Speicherplatine galvanisch zu verbinden.

Die 4-kByte-Blöcke werden dann dezentral mit Hilfe der angegebenen Chipselektschaltungen in die gewünschte Größe unterteilt.

In der Schaltung gemäß Bild 3.1 wird der Eingang G1 des *MH 74154* als notwendige logische Verknüpfung der Speicherselektierung mit dem $\overline{\text{MREQ}}$-Signal der CPU verwendet. Den Eingang G2 nutzt man zum Deselektieren des gesamten Speicherbereichs (G2 = H). Der Grund für diese Maßnahme wird im Zusammenhang mit der Startlogik erläutert.

Tabelle 3.1. Zentrale Speicherselektierung

Ausgang *MH 74154*	decodierter Speicherbereich (HEXA)	selektierter Speicherblock (kByte)
0	0000-0FFF	0- 4
1	1000-1FFF	4- 8
2	2000-2FFF	8-12
.	.	.
.	.	.
.	.	.
15	F000-FFFF	60-64

3.3.2. I/O-Portselektierung

Der Datenverkehr der CPU mit peripheren Geräten wird, sofern diese nicht wie gewöhnliche Speicherbereiche ansprechbar sind, über I/O-Ports (Ein-/Ausgabe-Tore) abgewickelt. Die Adressierung der I/O-Ports ergibt sich aus dem speziellen Verhalten der CPU *U880*. Diese liefert während eines I/O-Zyklus ($\overline{IORQ}$ = L) die 8 niederwertigen Adreßbit A0 ... A7 als I/O-Portadresse. Ob es sich um einen Eingabe- (Portlesen) oder Ausgabezyklus (Portschreiben) handelt, richtet sich nach der gleichzeitigen Aktivität des entsprechenden CPU-Signals $\overline{RD}$ oder $\overline{WR}$ (siehe [1], [2]).

Mit den 8 zur Verfügung stehenden Adreßbit A0 ... A7 können also maximal 256 I/O-Ports durch die CPU angesprochen werden. Die I/O-Ports werden im vorliegenden System folgendermaßen selektiert:

Als Decoderschaltkreis verwendet man wieder den von der Speicherselektierung bekannten 1-aus-16-Decoder *MH 74154*. Die Adreßbit AB0, AB1 werden im allgemeinen für spezielle Selektierungsfunktionen im Zusammenhang mit den I/O-Schaltkreisen PIO *U855* bzw. SIO *U856* benötigt (siehe Abschnitte 7. und 8.) und sind damit bereits belegt. Aus den in der Wertigkeit nächstfolgenden Adressen AB2, AB3, AB4, AB5 wird das I/O-Portselektsignal mit dem *MH 74154*, wie in Tabelle 3.2. angegeben, decodiert. Analog zur Speicherselektierung wird dabei die notwendige logische Verknüpfung mit dem Signal $\overline{IORQ}$ über den Toreingang G1 erreicht.

Damit verfügt man über 16 (×4) I/O-Portadressen. Die Unterscheidung der an jedem Ausgang vorliegenden 4 Einzeladressen ist

Tabelle 3.2. I/O-Portselektierung

Ausgang *MH 74154*	decodierte I/O-Ports (HEXA)
0	00, 01, 02, 03
1	04, 05, 06, 07
2	08, 09, 0A, 0B
.	.
.	.
.	.
15	3C, 3D, 3E, 3F

mit zusätzlichem Decodieraufwand (jeweils 1 aus 4) möglich. Für das System sind 16 I/O-Ports ausreichend, zumal bei Verwendung der systemspezifischen I/O-Schaltkreise PIO *U855* bzw. SIO *U856* über jedes der 16 Ports weitere schaltkreisinterne I/O-Kanäle angesprochen werden können (Port B/$\overline{A}$ Select über CPU-Adresse A0).
Ein gewünschtes I/O-Port wird der jeweiligen peripheren Einheit wieder durch Verbinden des entsprechenden Ausgangs des *MH 74154* mit dem Chipselekteingang der peripheriespezifischen Interfacebaugruppe (z. B. Chip-Enable-Eingang beim PIO- oder SIO-Schaltkreis) zugeordnet.

3.3.3. Startlogik

Der Mikroprozessor *U 880* kann grundsätzlich über $\overline{\text{RESET}}$ oder über Interrupt gestartet werden (siehe [1]). Für den Urstart des Systems verwendet man üblicherweise das bereits erwähnte „Power-On-RESET". Hierbei beginnt die CPU die Befehlsbearbeitung bei der Speicherzelle 0. Aus diesem Grund ist es häufig so, daß der Beginn des Betriebssystems des Mikrocomputers (z. B. eines Monitorprogramms) auf die Speicheradresse 0 gelegt wird. Diese Methode ist mit einigen Nachteilen verbunden, die den Ausschlag für die Wahl einer anderen Lösung gaben. Die genannten Nachteile sind vor allem softwarebedingt und ergeben sich daraus, daß im oben beschriebenen Fall (d. h. Monitorbeginn bei Adresse 0) ab Adresse 0 kein frei verfügbarer Speicherbereich vorhanden ist. Dies soll im folgenden kurz erläutert werden.
Für Mikrocomputer auf der Basis des *U 880* ist heute ein breites Softwareangebot bekannt, das von Assembler- und Disassemblerprogrammen bis zu Interpreter- bzw. Compilerprogrammen für höhere Programmiersprachen (z. B. BASIC, PASCAL, FORTRAN usw.) reicht. In vielen Fällen läuft diese Standardsoftware ab Adresse 0. Ein Umschreiben bewährter Programme für andere Speicherbereiche ist mit großem Aufwand und entsprechend vielen Fehlermöglichkeiten verbunden. Als weiterer Aspekt ist zu beachten, daß einige der genannten Programme nur auf RAM-Speichern lauffähig sind, da sich in diesem Fall die notwendigen Arbeitszellen innerhalb des jeweiligen Programmspeicherbereichs befinden (also nicht in andere RAM-Bereiche ausgelagert sind).

Im Anfangsbereich des Adreßraums des *U880* befinden sich einige Adressen mit besonderer Bedeutung (00H, 08H, 10H ... 38H, 66H: Restart bzw. Interruptmode 1 und $\overline{\text{NMI}}$), die möglichst allgemein verfügbar sein sollten.
Aus den genannten Gründen wurde im System die Voraussetzung dafür geschaffen, daß – beginnend bei der Adresse 0 – ein möglichst großer, zusammenhängender RAM-Bereich realisiert werden kann. In diesem Zusammenhang müssen noch ein paar Überlegungen zur Speicherbereichsaufteilung angestellt werden:
Man benötigt für das System unbedingt ein Monitorprogramm (siehe Abschnitt 9.), welches das Arbeiten auf Maschinensprachniveau sowie die Handhabung aller weiteren Software ermöglicht. Als wichtiges, den Dialog zwischen Mensch und Computer unterstützendes peripheres Gerät wird ein Fernsehinterface benötigt (siehe Abschnitt 5.). Wie sich zeigen wird, kann man dieses durch den Computer wie einen normalen RAM-Speicherbereich ansprechen.
Im Ergebnis der Überlegungen kommt man zu der im Bild 3.2 angegebenen Speicherbereichsaufteilung.

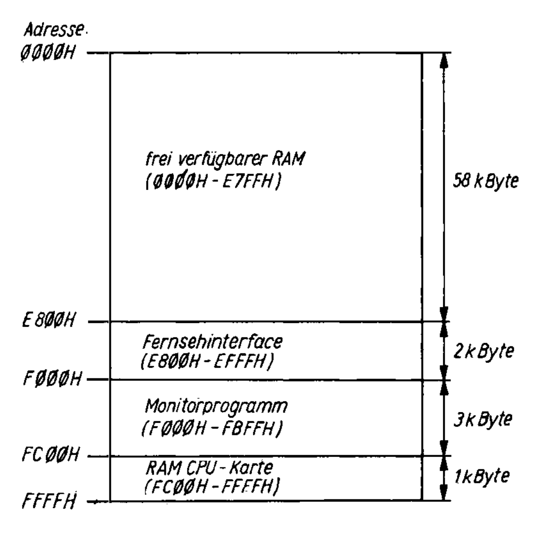


Bild 3.2 Speicherbereichsaufteilung

Für den Bildwiederholspeicher des Fernsehinterface werden 2-kByte im Adreßbereich E800H ... EFFFH reserviert. Das in Abschnitt 9. beschriebene Monitorprogramm wird im Bereich F000H FC00H untergebracht. In den bis zum Ende des Adreßraums verbleibenden 1-kByte-Bereich paßt genau der Arbeitsspeicher (1-kByte-RAM) der CPU-Platine.
Damit sind die oberen 6 kByte des 64-kByte-Adreßbereichs des Computers belegt, und es steht ein zusammenhängender Adreßraum von 58 kByte (0000H ... E7FFH) für Anwenderprogramme zur freien Verfügung.
Bei der gewählten Speicheraufteilung muß der Beginn der Programmbearbeitung bei der vereinbarten Startadresse des Monitorprogramms F000H gewährleistet werden. Hierfür gibt es hard- und softwareorientierte Möglichkeiten. Beispielsweise kann man den Monitor einschließlich einer einfachen Laderoutine ab Adresse 0 auf PROM unterbringen. Die Laderoutine dupliziert dann nach Netzeinschalten den Monitor zur Adresse F000H, wo in diesem Fall natürlich ein RAM-Speicher zur Verfügung stehen muß. Nach Beendigung dieses Vorgangs wird der PROM deselektiert, damit ab Adresse 0 ein frei verfügbarer Speicher vorhanden ist. Auf weitere Einzelheiten dieser Variante soll nicht eingegangen werden, da in dieser Broschüre die in Bild 3.1 angegebene einfache Hardwarelösung verwendet wird.
Die Funktionsweise der Startlogik soll an Hand des Impulsbildes (Bild 3.3) erläutert werden.
Mit dem Power-On-RESET wird das RS-Flip-Flop (D 6.2, D 6.3) gesetzt, alle Datenleitungen werden über Open-Collektor-Schaltkreise D 7, D 8 auf L gezogen. Das bedeutet, daß die CPU – beginnend bei der Adresse 0 – mit jedem Befehlsholezyklus (M1-Zyklus) den Operationscode 00H, d. h. den Befehl NOP, einliest. Mit NOP führt die CPU bekanntlich einen Leerzyklus aus. In dieser Weise wird der Adreßbereich 0000H ... EFFFH durchlaufen. Bei Erreichen der Adresse F000H (unter logischer Verknüpfung mit $\overline{\text{MREQ}}$) erscheint am Ausgang von Gatter D 9 L-Potential, das RS-Flip-Flop kippt zurück, und der Datenbus wird für normale Befehlsbearbeitung freigegeben.
Um einen Kurzschluß von Speicherausgängen beim Hochstartvorgang zu vermeiden (DB0 ... DB7 = L!), benutzt man das Signal MEMDES am Ausgang von Gatter D 2.4 dazu, über den Toreingang G 2 der zentralen Speicherselektierung sämtliche angeschlos-

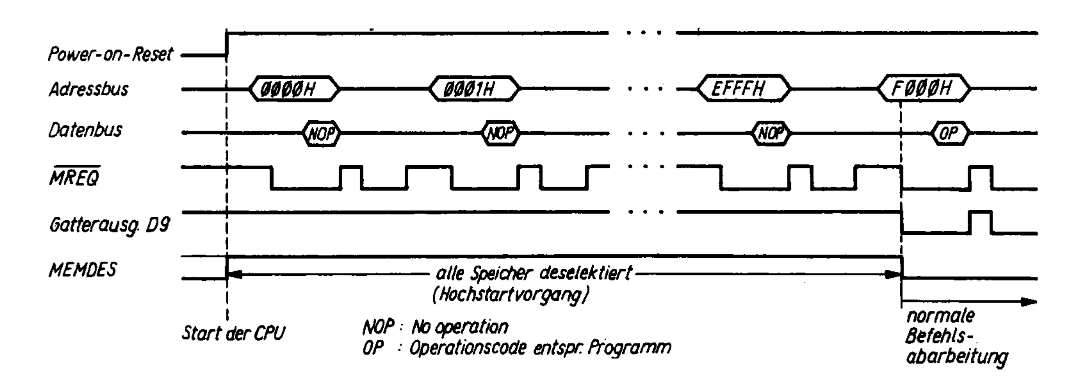


Bild 3.3 Impulsdiagramm Startlogik

senen Speicherschaltkreise zu deselektieren (siehe Abschnitt 3.3.1.).

3.4. Aufbau und Inbetriebnahme

Aufbau und Inbetriebnahme der CPU-Baugruppe sollen mit einfachen, auch dem Amateur zugänglichen Hilfsmitteln realisiert werden. Es müßte jedoch ein Oszilloskop – möglichst als Zweistrahlenausführung – für diesen Zweck verfügbar sein.
Die CPU-Platine kann man dann wie folgt systematisch bestücken und erproben:
1. Zunächst werden nur der Quarzoszillator einschließlich 1 : 4-Untersetzer und Puffergatter sowie die $\overline{\text{RESET}}$-Logik auf der Leiterplatte bestückt. CPU und EPROM sind nicht auf die Fassungen gesteckt. Außer den genannten sind keine weiteren Schaltkreise eingelötet.
Mit dem Oszilloskop ist am Oszillatorausgang die Frequenz von 10 MHz und am Pin 6 der CPU-Fassung die Rechteckspannung mit der Frequenz von 2,5 MHz zu kontrollieren.
2. Danach bestückt man die Startlogik mit den Schaltkreisen D 6, D 7, D 8, D 9 und den Widerständen R 9 ... R 16. Die Funktion der RESET-Schaltung und der Startlogik kann nun zusammen kontrolliert werden.
Nach Einschalten der Betriebsspannung + 5 V bewirkt das RESET-Signal am Eingang von Gatter D 6.2 das Setzen des RS-Flip-Flop, alle Ausgänge der Open-Collektor-Gatter D 7, D 8 müssen bei ordnungsgemäßer Funktion L-Signal führen. Beim Anlegen eines H-Signals an die Eingänge für AB 12 ... AB 15 von D 9 und eines L-Signals an den Gattereingang D 6.1 muß das RS-Flip-Flop zurückkippen, und die Gatterausgänge von D 7, D 8 müssen auf H liegen.
3. Einlöten der Bustreiber *8212* und *8216* sowie des Schaltkreises D 4. Anschließend wird der Flip-Flop-Eingang D 6.3 vom Gatterausgang D 9 getrennt und auf H gelegt. Damit ist ein Zurückkippen des RS-Flip-Flop nach Erreichen der Adresse F000H beim Startvorgang verhindert. Zuletzt steckt man die CPU auf die Fassung.
Nach Einschalten der Betriebsspannung und dem folgenden $\overline{\text{RESET}}$ werden alle Datenleitungen DB0 ... DB7 auf L-Potential gezogen. Durch die Blockierung des RS-Flip-Flop ist dieser Zustand permanent vorhanden.

Infolge dieser Maßnahme liest die CPU ausschließlich NOP-Befehle ein, und der gesamte Adreßbereich wird zyklisch durchlaufen. Mit dem Oszilloskop kontrolliert man jetzt die Adreßleitungen AB0 ... AB15. Die an AB0 anliegende Pulsfolge weist die höchste Frequenz auf, die an AB15 die niedrigste. In ansteigender Reihenfolge der Adressen muß an der jeweiligen Adreßlinie die halbe Frequenz der vorhergehenden anliegen. Obwohl die Impulsfolge an den einzelnen Adreßausgängen wegen des Refresh-Zyklus unterbrochen ist, muß der beschriebene Sachverhalt deutlich auf dem Oszilloskop erkennbar sein. Auch an den Steuersignalausgängen $\overline{\text{RD}}$, $\overline{\text{MREQ}}$ und $\overline{\text{M1}}$ muß eine Impulsfolge anstehen.

4. Nun wird noch die Chipselektierung für die Speicherschaltkreise der CPU-Baugruppe überprüft.

Dazu lötet man als nächstes die Schaltkreise D 17 und D 5 ein. Die Datenleitungen bleiben, wie unter 3. beschrieben, auf L-Potential. Der Blockselekteingang C des *MH 7442* (D 17) wird vom Ausgang des Gatters D 9 getrennt und auf Masse gelegt. Nach dem Einschalten der Betriebsspannung muß ein Impuls an den $\overline{\text{CS}}$-Eingängen der EPROM-Speicher (Pin 20) bzw. des RAM-Blocks (Pin 13) anliegen. Mit dem Zweistrahloszilloskop kann überprüft werden, ob die 4 $\overline{\text{CS}}$-Signale – wie es notwendig ist – zeitlich versetzt auftreten. Man darf natürlich nur jeweils 1 Speicherschaltkreis (bzw. RAM-Block) gleichzeitig selektieren.

5. Zum Abschluß werden alle noch fehlenden Schaltkreise und Bauelemente eingelötet. Die Blockierung des RS-Flip-Flop der Startlogik wird aufgehoben, und alle Verbindungen gemäß Schaltung in Bild 3.1 werden wieder hergestellt.

Wer jetzt über die Möglichkeit verfügt, einen PROM programmieren zu lassen (was er später zur Inbetriebnahme des Monitor-Programms ohnehin muß), kann die Ein-/Ausgabefunktion noch testen.

Dazu programmiert man einen EPPROM mit:

```
M1:   IN A
      OUT A
      INC C
      JR M1
```

Der EPROM wird auf den Steckplatz 1 der CPU-Platine gesteckt und das kleine Programm mit Power-On-RESET und Startlogik bei F000H gestartet. Auf den Steuerleitungen $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ und

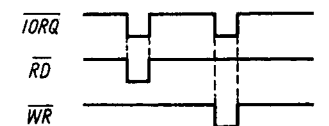


Bild 3.4 Impulsdiagramm Ein-/Ausgabefunktion

$\overline{\text{WR}}$ muß der in Bild 3.4 dargestellte Impulsverlauf nachweisbar sein.
Die Ausgänge der I/0-Portdecodierung müssen I/0-Selektsignale aufweisen.
Diesen Zeitverlauf kontrolliert man mit dem Zweistrahloszilloskop.
Der Test ermöglicht gleichzeitig noch einmal den Nachweis für die einwandfreie Funktion der Startlogik.
Nach positivem Abschluß der Inbetriebnahme gemäß 1. bis 5. sind wesentliche Funktionen der CPU-Baugruppe überprüft. Man kann jetzt mit hoher Wahrscheinlichkeit davon ausgehen, daß ihre ordnungsgemäße Arbeitsweise abgesichert ist. Eine Gesamtüberprüfung stellt dann noch einmal die später folgende Inbetriebnahme des Monitorprogramms im Zusammenhang mit den peripheren Baugruppen dar.
Es soll hier aber auch nicht unerwähnt bleiben, daß manchmal bestimmte versteckte Fehler erst nach längerer Betriebszeit entdeckt werden.

4. Speichererweiterung

Wie bereits erwähnt, stellt die in Abschnitt 3. beschriebene CPU-Baugruppe einen in sich funktionsfähigen Einplatinenrechner dar. Zu einem leistungsfähigen Mikrocomputersystem gehört selbstverständlich mehr, vor allem eine ausreichend aufgerüstete Speicherbaugruppe.
Im Zusammenhang mit der Startlogik wurde darauf hingewiesen, daß im vorliegenden Systemkonzept ein möglichst großer, zusammenhängender RAM-Bereich bevorzugt wird. Voraussetzung für die Abarbeitung häufig benötigter Programme im RAM-Bereich ist aber ihr vorheriges Laden von einem externen Speichermedium (Kassette, Lochband, Floppy-Disk) bzw. von einem Festwertspeicher (PROM bzw. EPROM). Es ist leicht einzusehen, daß es nicht möglich ist, beispielsweise einen 8 ... 12 kByte langen BASIC-Interpreter jedesmal vor einer gewünschten Benutzung über die Tastatur einzugeben.
Weiterhin ist es sinnvoll, ein häufig benötigtes Programm, wie z. B. den erwähnten Monitor, direkt auf PROM-Speicher laufen zu lassen. Voraussetzung ist natürlich, daß er dafür geeignet ist.
Im folgenden werden die Schaltungen für die zur Speichererweiterung notwendigen ROM- und RAM-Platinen angegeben und diskutiert. Als RAM-Speicher werden je eine Schaltungsvariante auf der Basis des statischen RAM *U 202* sowie des für einen effektiven Speicherausbau günstigen dynamischen RAM *U 256 (4116)* vorgestellt.
An dieser Stelle ist es erforderlich, auf das Problem der Pufferung der Busleitungen auf Speicher- und Peripheriebaugruppen kurz einzugehen.
Der Datenverkehr im Computer ist grundsätzlich bidirektional. Das bedeutet also, daß alle Datenquellen (außer der CPU auch Speicher und Peripherie) in der Lage sein müssen, den Datenbus mit allen angeschlossenen Systemkomponenten hinsichtlich Lastfaktor und kapazitiver Belastung zu treiben.
Aus der Sicht der CPU ist dies durch die eingesetzten Treiberschaltkreise abgesichert. Für Speicher- und Peripheriebaugruppen in größeren Mikrocomputersystemen können sich aber in diesem

Zusammenhang Probleme ergeben. Besonders beachten muß man die Kapazitäten der Ein- und Ausgänge von Speicherschaltkreisen, die sich durch die übliche Parallelschaltung der Adreß- und Datenleitungen entsprechend vergrößern.
Aus den genannten Gründen ist es bei allgemein verwendbaren OEM-Computern wie z. B. *K 1520* (OEM: *Original Equipment Manufactures*) sowie in großen Systemen üblich und notwendig, die Leitungen auf der jeweiligen Speicher- bzw. Peripheriebaugruppe zu puffern. Aus Gründen des Schaltkreisbedarfs ist es im vorliegenden System vertretbar, auf diese Pufferung zu verzichten. Dabei muß man darauf achten, daß die Belastung des Busses kleiner als eine TTL-Last bleibt und sich die oben erwähnten Lastkapazitäten in annehmbaren Grenzen halten.
Wer über ausreichend Treiberschaltkreise verfügt, sollte die Busleitungen auf den Speicherplatinen (z. B. PROM mit *8212*, RAM mit *8216*) bzw. Peripheriebaugruppen jedoch puffern.

4.1. ROM-Speicher

Bild 4.1 zeigt die Schaltung für eine 8-kByte-Prom-Leiterplatte auf der Basis des Schaltkreises *U 555 (2708)*.
Mit AB0 ... AB9 werden die 1k 8-Bit-Speicherzellen auf dem jeweiligen Chip adressiert. Das Chipselektsignal für die 8 auf der Leiterplatte plazierten PROM wird durch 1-aus-8-Decodierung der Adressen AB10, AB11, AB12 sowie des in Abschnitt 3.3.1. erläuterten Blockselektsignals der zentralen Speicherselektierung erzeugt. Für diesen Zweck nutzt man den 1-aus-10-Decoder *MH 7442*. Dem Problem direkt angepaßt ist natürlich der eventuell schwieriger beschaffbare 1-aus-8-Decoder *8205*.
Die Pufferung der Datenausgänge bleibt, wie oben ausgeführt, dem einzelnen überlassen. Falls sie gepuffert werden sollen, eignet sich dazu der Schaltkreis *8212*. Dieser Schaltkreis bietet sich deshalb an, weil vom PROM-Speicher selbstverständlich nur Daten gelesen werden können (der Fall des PROM-Programmierens ist in diesem Zusammenhang nicht von Interesse) und damit nur Datenfluß in der einen Richtung PROM → CPU möglich ist.
Mit H-Pegel am STR-Eingang wird die 8-Bit-Information der jeweiligen Speicherzelle des gerade selektierten PROM in den D-Latches des *8212* gespeichert. Wenn $\overline{DS1}$ = L (entsprechend

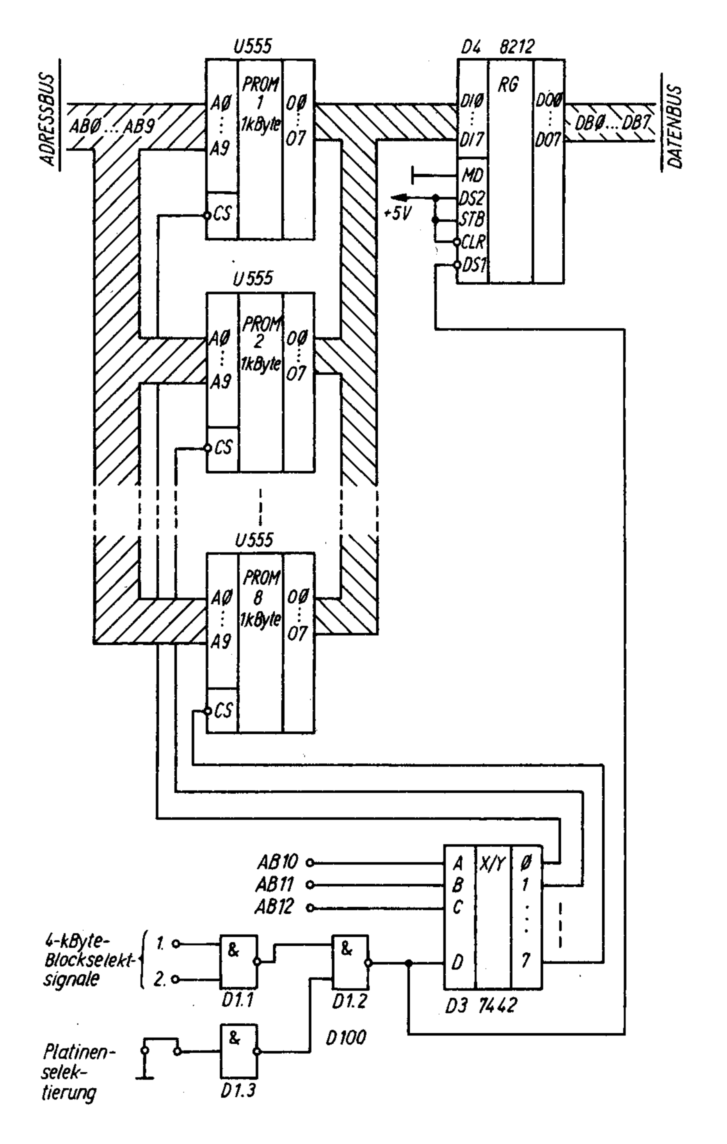


Bild 4.1 8-kByte-PROM-Platine mit *U 555*

dem Blockselektsignal) und DS2 = H, liegt die Information an den Tri-State-Ausgängen des *8212* und damit gepuffert am Datenbus.
Die zentrale Speicherselektierung liefert 4-kByte-Blockselektsignale, von denen für eine 8-kByte-Platine natürlich 2 Stück benötigt werden. Im allgemeinen schließt man 2 adressenmäßig benachbarte Blockselektsignale an die Eingänge von Gatter D1.1 an und erhält damit einen durchgängigen 8-kByte-Prom-Speicher. Wenn es sinnvoll erscheint, kann man diesen Speicher auch in zwei 4-kByte-Blöcke unterteilen.
Die Selektierung bzw. Deselektierung der gesamten Speicherplatine wird über Gatter D1.3 vorgenommen. Liegt der Gattereingang auf Masse (siehe Bild 4.1), ist die Platine selektiert.
Nachdem die Speichereinheit auf der Basis von 1k × 8-organisierten PROM vorgestellt wurde, seien noch einige Hinweise gegeben, die es gestatten, die Platine so zu gestalten, daß sie auch für künftig verfügbare Speicherschaltkreise mit entsprechend größeren Speicherdichten verwendbar sind. Dies wurde möglich, weil es für alle nach dem Byte-wide-Prinzip (8 Bit Speicherbreite) organisierten Speichertypen eine standardisierte Pin-Belegung gibt. Diese Pin-Belegung gilt für die PROM-Typen *U555 (2708), 2716, 2732, 2764* und ist dem Anhang zu entnehmen. Man muß beachten, daß der 8-kByte-Speichertyp (z. B. *2764*) 28polig ausgeführt ist.
International existieren bereits RAM-Speicher [z. B. *6116* (Hitachi), *4118* (MOSTEK)], die zu den genannten PROM-Typen pinkompatibel sind. Damit wäre es sogar möglich, eine entsprechend gestaltete Leiterplatte wahlweise als PROM- und/oder als RAM-Platine zu verwenden. Beim Einsatz von RAM-Schaltkreisen ist es natürlich notwendig, die CPU-Signale $\overline{RD}$ und $\overline{WR}$ mit auf die Leiterplatte zu führen und für die Lese/Schreib-Umschaltung heranzuziehen.
Bild 4.2 zeigt die Schaltung einschließlich Verdrahtungsprinzip für eine universell verwendbare Speicherplatine.
Die Adreßeingänge A10, A11, A12 sowie die $\overline{CS}$-Eingänge der Speicherschaltkreise werden ebenso wie die Eingänge des Chipselektdecoders *MH 7442* als unbelegte Wickel- oder Lötanschlüsse auf der Platine ausgeführt. Sie können damit durch Drahtbrücken so beschaltet werden, wie es der jeweilige Speichertyp erfordert.
Die Tabelle 4.1. enthält die notwendige Adreßbelegung für vier 8-Bit-organisierte Speichertypen. Die Verdrahtung für den 1-kByte-Speicher (z. B. *U555*) entspricht der in Bild 4.1, wobei man 2 Ein-

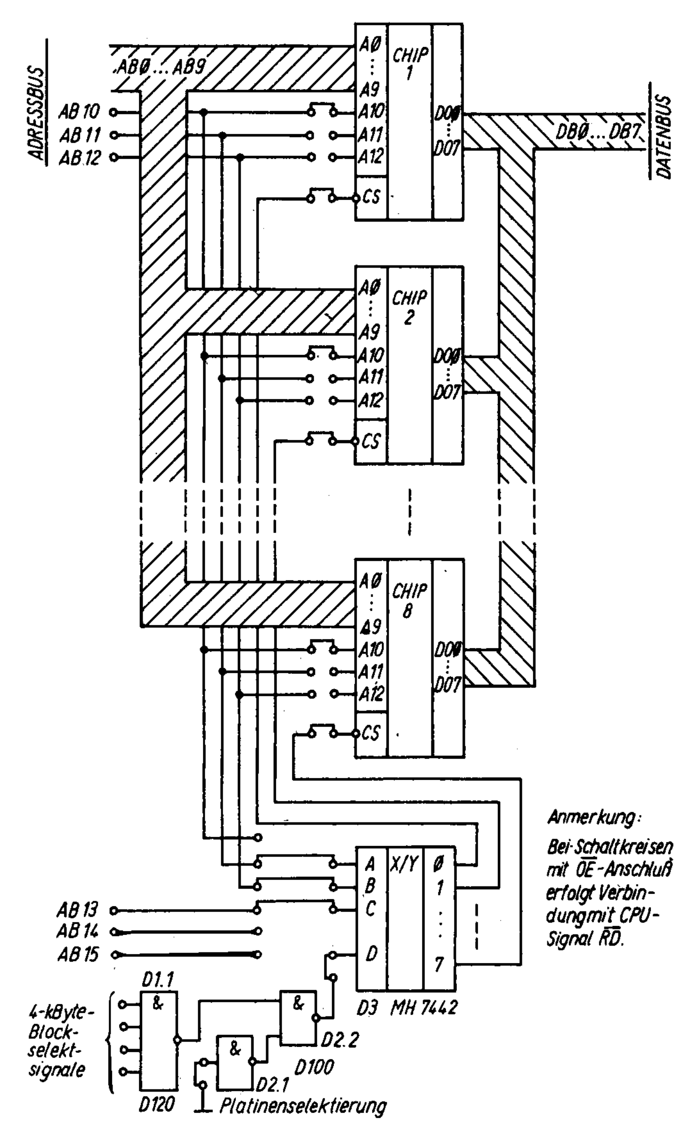


Bild 4.2 Universell verwendbare Platine für byte-wide-organisierte Speichertypen

Tabelle 4.1. Adreßbelegung für universell verwendbare Speicherplatine

Adreßbus	SPEICHERTYP			
	U 555 (1 kByte)	2716 (2 kByte)	2732 (4 kByte)	2764 (8 kByte)
AB0...AB9	A0...A9	A0...A9	A0...A9	A0...A9
AB10	A } Ein-	A10	A10	A10
AB11	B } gänge	A } Ein-	A11	A11
AB12	C } *MH7442*	B } gänge	A } Ein-	A12
AB13	/	C } *MH7442*	B } gänge	A } Ein-
AB14	/	/	C } *MH7442*	B } gänge
AB15	/	/	/	C } *MH7442*

gänge des Gatters D 1.1 in Bild 4.2 auf H-Pegel legen muß. Beim 2-kByte-Speichertyp (z. B. *2716*) sind 4 (im allgemeinen benachbarte) 4-kByte-Blockselektsignale der zentralen Speicherselektierung an die Eingänge von Gatter D 1.1 anzuschließen. Für den 4-kByte-Speichertyp (z. B. *2732*) wird der Chipselektdecoder nicht benötigt. Die Ausgänge des *MH 74154* der zentralen Speicherselektierung werden direkt mit den $\overline{CS}$-Eingängen der 8 Speicherschaltkreise verbunden. Für den 8-kByte-Speicher (z. B. *2764*) ist die zentrale Speicherselektierung also gegenstandslos, da 8 Chips in diesem Fall den gesamten Adreßbereich der CPU *U 880* ausfüllen. Dementsprechend ist der Eingang D des *MH 7442* auf Masse zu legen, wobei der Gatterausgang von D 2.2 natürlich vorher abgetrennt werden muß.

4.2. RAM-Speicher

4.2.1. Statischer RAM-Speicher

Das Schaltungsprinzip für eine Speicherplatine auf der Basis des statischen RAM *U 202* wurde bereits im Zusammenhang mit dem auf der CPU-Baugruppe untergebrachten Speicher in Abschnitt 3.2. (Bild 3.1) beschrieben. Der Aufbau einer 4-kByte-RAM-Platine, bei der der Treiberschaltkreis *8212* verwendet wird, ist damit sinngemäß möglich. Die Chipselektierung, d. h. in diesem Fall das Selektieren der vier 1-kByte-RAM-Blöcke, sowie die Lese/

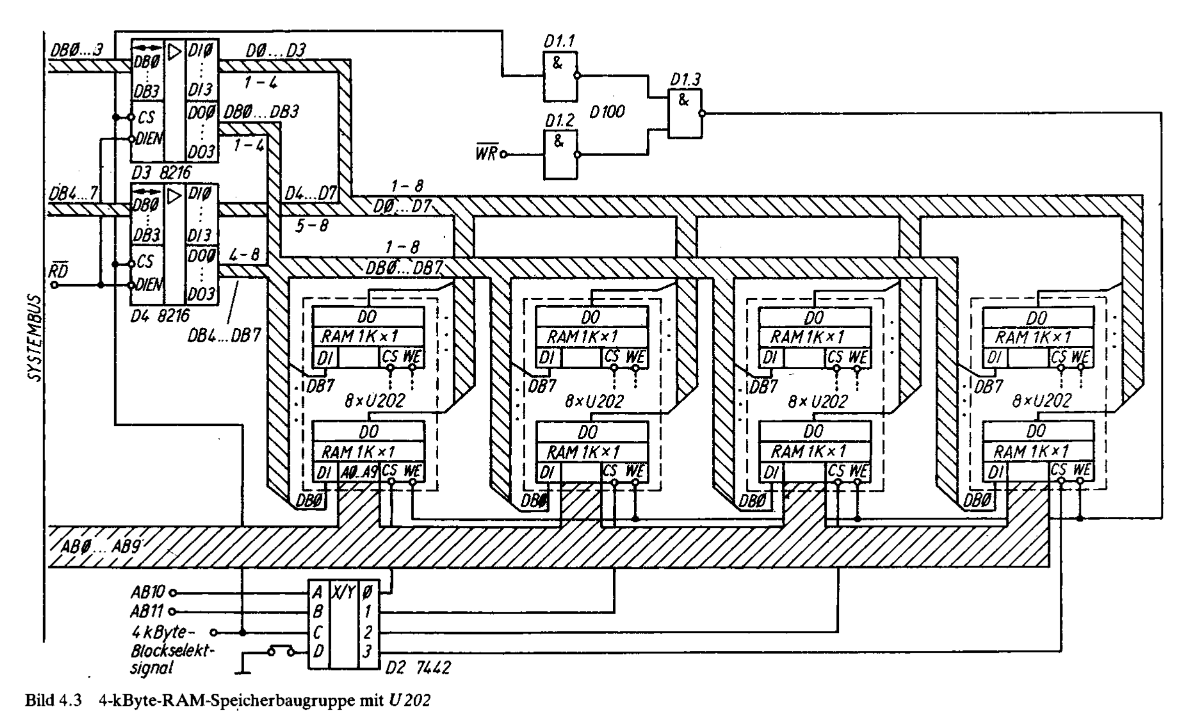
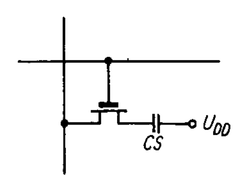


Bild 4.3 4-kByte-RAM-Speicherbaugruppe mit *U* 202

Schreib-Umschaltung sind identisch. Die 3 PROM ersetzt man durch 1-kByte-RAM-Blöcke und faßt die Datenausgänge DO0 ... DO7 entsprechend ihrer Wertigkeit zusammen. Nach dem Puffern durch den *8212* werden sie auf den Datenbus geschaltet.
Bild 4.3 zeigt die Datenpufferung mit 2 bidirektionalen Treiberschaltkreisen. Wie ebenfalls bereits in Abschnitt 3.3. erwähnt wurde, ist (bei Vorhandensein der benötigten Treiber *8216*) dieser Variante der Vorzug zu geben.

4.2.2. Dynamischer RAM-Speicher

Dynamische RAM haben gegenüber statischen RAM einige Vorteile: So werden in einem Schaltkreis weitaus mehr Speicherzellen vereinigt, die Leistungsaufnahme ist bedeutend geringer, und der Preis je Bit ist kleiner.
Bisher wurden nur statische Speicherschaltkreise verwendet, denn sie lassen sich leicht handhaben, und sie benötigen keine externe Steuerung. Da es kaum Timing-Probleme gibt, läßt sich ein Computer mit statischen RAM wesentlich einfacher in Betrieb nehmen. Die Autoren empfehlen den Bau dieser dynamischen RAM-Karte deshalb als Erweiterung. Die Inbetriebnahme ist für den Amateur sonst eventuell zu schwierig.
Im Gegensatz zu statischen RAM, in denen bekanntlich Flip-Flop die Informationen speichern, enthalten die modernen dynamischen RAM Eintransistorzellen (Bild 4.4a). Gespeichert wird in der Kapazität C_S. Diese Speicherzellen gestatten hohe Integrationsdichten, d. h., es lassen sich sehr viele Speicherplätze auf einem Chip unterbringen.
Für die dynamische RAM-Karte wurde der 16384 × 1-Bit-RAM *U256 (4116)* verwendet. Die Funktionsweise eines dynamischen 16-kBit-RAM soll kurz an Hand von Bild 4.4b erläutert werden. Die einzelnen Speicherzellen sind in einer Matrix von 128 Zeilen (row) und 2 × 64 Spalten (column) angeordnet. Die 14-Bit-Adresse wird dem RAM zu je 7 Bit multiplex als Zeilen- und Spaltenadresse zugeführt. Die beiden Adressen werden in zwei 7-Bit-Latches übernommen: Die Zeilenadresse durch einen $\overline{RAS}$-Impuls (*row address strobe*), die Spaltenadresse durch einen $\overline{CAS}$-Impuls (*column address strobe*). Wie bereits erwähnt, wird die Information in Kapazitäten gespeichert, die sich durch Leckströme langsam entla-



WR
RAS
CAS
Takt-generator 1
Takt-generator 2
Takt-generator 3
&
&
Latch
DI
Buffer
DO
7 Bit-Latch
Decoder
Speichermatrix
128 Leseverstärker
Speichermatrix
Bus-sel.
A0 … A6
1
64
65
128
1 … 64
Decoder
7 Bit-Latch

b)

Bild 4.4 a – dynamischer RAM *U 256 (4116)*, Eintransistorspeicherzelle, b – Übersichtsplan

den. Um einen Informationsverlust zu vermeiden, muß jede Zeile mindestens alle 2 ms aufgefrischt werden (*refresh*). Das ist übrigens eine beachtlich lange Zeit, wenn man bedenkt, daß die Kapazität einer Speicherzelle nur etwa 0,04 pF beträgt!
Beim Lesen und Auffrischen verstärken 128 Leseverstärker die Ladung der Kapazitäten. Die Leseverstärker sorgen für ein Nachladen der Kapazitäten und beschreiben während des Schreibvorgangs ($\overline{WE}$ = L) die Zellen neu.
Der dynamische 16-kBit-RAM ermöglicht verschiedene Betriebsarten. Gesteuert wird über die Eingänge $\overline{RAS}$, $\overline{CAS}$ und $\overline{WE}$. Bild 4.5 zeigt Impulsdiagramme für die verwendeten Betriebsarten. Die angegebenen Zeiten sind typische Werte (abhängig von der Zugriffszeit und vom Hersteller).
Die in Bild 4.6a gezeigte Schaltung realisiert die Ansteuerung für maximal 64-kByte-RAM. Der Decoder D1 (*8205*) wählt einen der 16-k-Blöcke aus. Der Schaltkreis D2 wird als Latch genutzt. Er verhindert Konflikte, die sich durch vorzeitigen Adreßwechsel ergeben können [2]. Ein Speicherzugriff ($\overline{MREQ}$ = L, $\overline{RFSH}$ = H) gibt den Decoder frei. Einer der Decoderausgänge wird aktiv (L-Pegel). Der entstehende Impuls steht hinter den Gattern D 4.1 ... D 4.4 als $\overline{RAS1}$, $\overline{RAS2}$, $\overline{RAS3}$ oder $\overline{RAS4}$ bereit. An den Adreßeingängen des Speicherschaltkreises liegen die 7 niederwertigen Adreßleitungen. Nachdem der RAM die Zeilenadresse übernommen hat, kann der Multiplexer (D8 und D9) umschalten und die 7 höherwertigen Adreßleitungen an den RAM legen. Liegt die Adresse sicher am RAM an, wird sie durch $\overline{CAS}$ übernommen. Die Verzögerungskette ist für das Timing verantwortlich. Um die Early-Write-Betriebsart zu realisieren, wird das $\overline{CAS}$-Signal nochmals verzögert (Gatter D6.2, D6.3, D7.3, D7.4). Das $\overline{CAS}$-Signal steuert außerdem die Tri-State-Treiber.
Während eines Refreshzyklus ($\overline{MREQ}$ = L, $\overline{RFSH}$ = L) ist der Decoder D1 inaktiv. Am Ausgang des Gatters D3.3 liegt L-Pegel, so daß über D4.1 – D4.4 alle RAS-Leitungen L-Pegel erhalten. Es wird ein „RAS-ONLY"-Zyklus ausgelöst (eine Zeile der Matrix wird aufgefrischt). Bild 4.7 soll die Vorgänge verdeutlichen.
Die kleine Logik in Bild 4.6b mit den Gattern D13.1, D13.2 und D5.2 hat die Aufgabe, die oberen 6 kByte der 64-kByte-RAM-Karte auszublenden. Der Bereich von E800H... FFFFH wird bereits von den Speichern auf der CPU-Karte und dem Bildwiederholspeicher belegt.

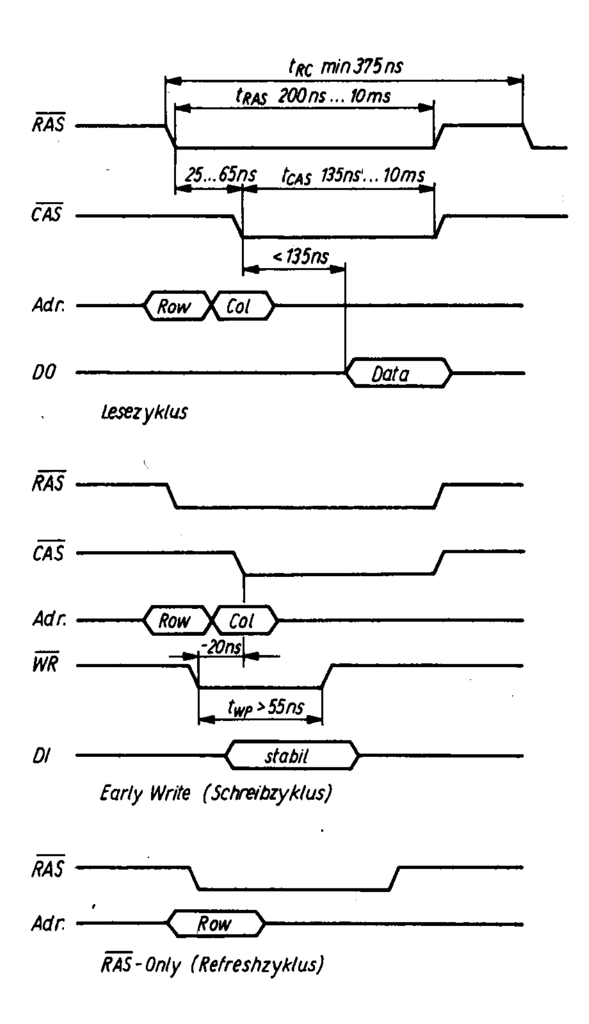
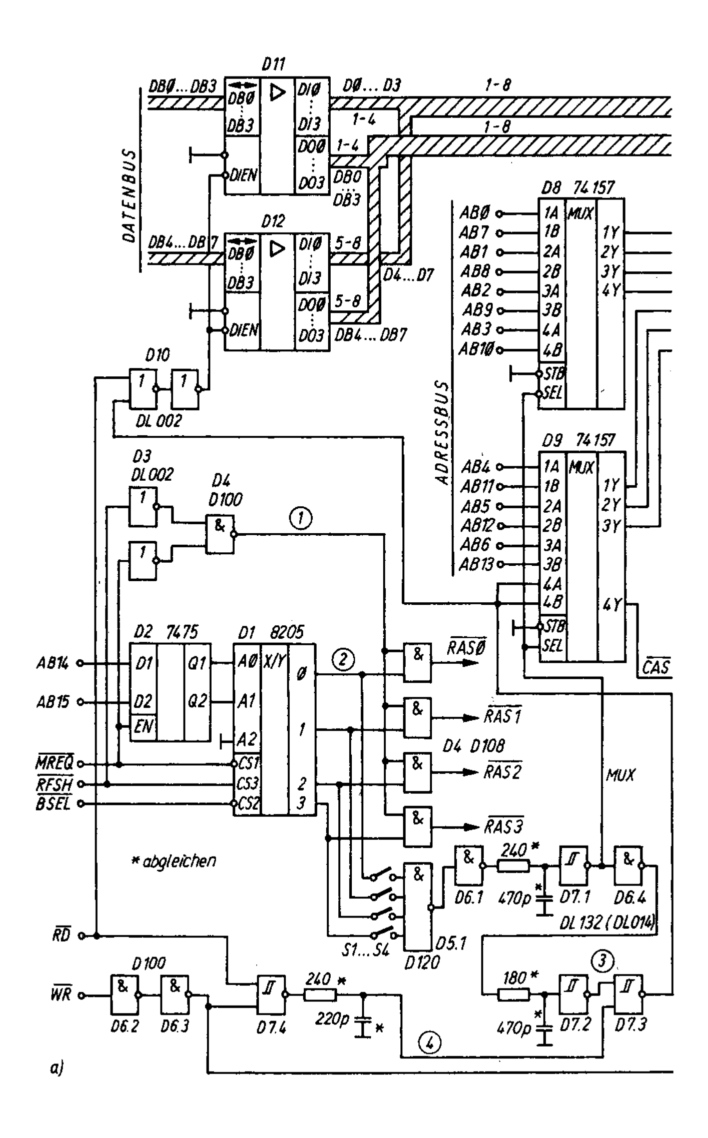
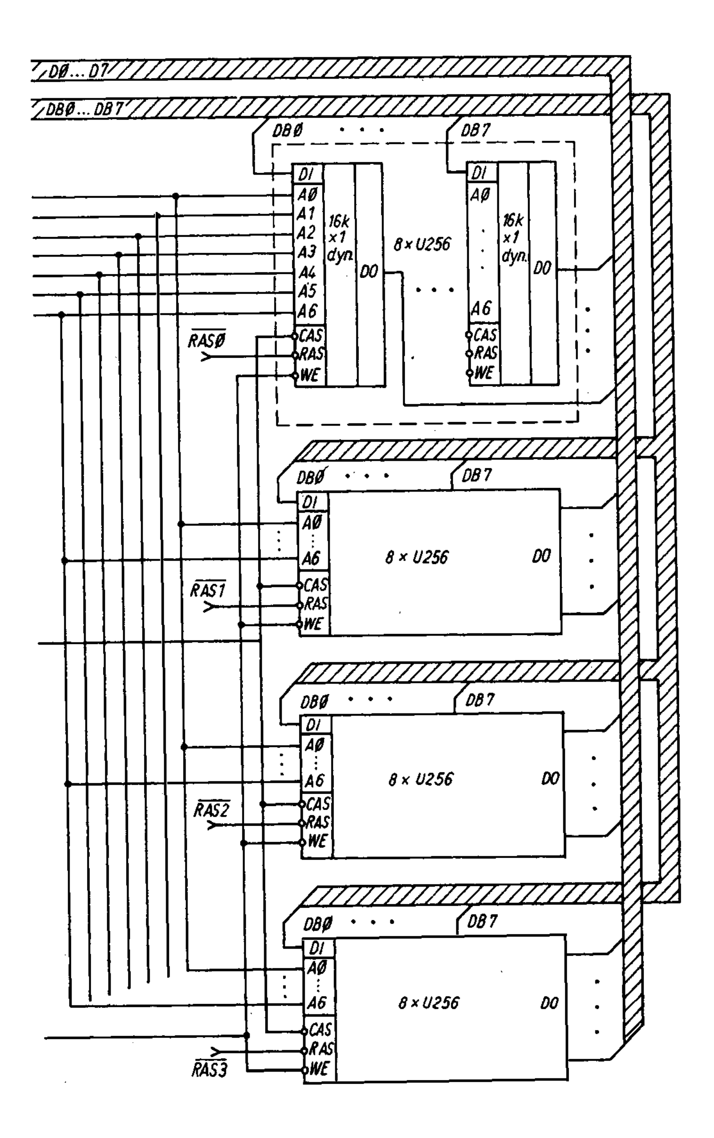


Bild 4.5 Impulsdiagramm der dynamischen RAM *U 256 (4116)*


D11
DBØ...DB3
DBØ
DB3
DIØ
DI3
DØ... D3
1-8
1-4
1-8
DOØ
DO3
1-4
DIEN
DB0
DB3
DATENBUS
D12
DB4...DB7
5-8
D4...D7
5-8
DB4...DB7
D10
DL 002
D8 74157
MUX
AB0
AB7
AB1
AB8
AB2
AB9
AB3
AB10
1A
1B
2A
2B
3A
3B
4A
4B
STB
SEL
1Y
2Y
3Y
4Y
ADRESSBUS
D9 74157
AB4
AB11
AB5
AB12
AB6
AB13
D3
DL002
D4
D100
D2 7475
D1 8205
X/Y
AB14
AB15
D1
D2
EN
Q1
Q2
A0
A1
A2
CS1
CS3
CS2
RAS0
RAS1
RAS2
RAS3
D4 D108
CAS
MUX
MREQ
RFSH
BSEL
* abgleichen
240 *
470p
D6.1
D7.1
D6.4
DL 132 (DL014)
D5.1
D120
S1...S4
RD
WR
D100
D6.2
D6.3
D7.4
240 *
220p
180 *
470p
D7.2
D7.3
a)


DØ ... D7
DBØ ... DB7
DBØ
DB7
DI
AØ
A1
A2
A3
A4
A5
A6
CAS
RAS
WE
16k
x1
dyn.
DO
8 × U256
RASØ
RAS1
RAS2
RAS3

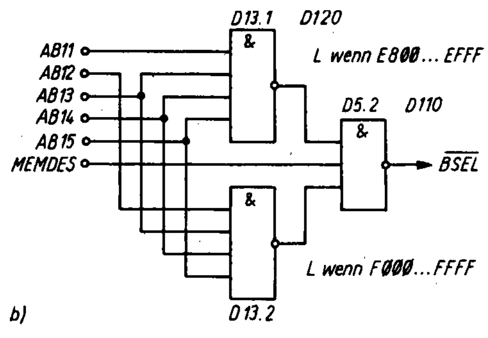


Bild 4.6 a – 64-kByte-RAM-Speicherbaugruppe mit *U 256 (4116)*, b – Logik zum Ausblenden des Speicherbereichs E800H . . . FFFFH

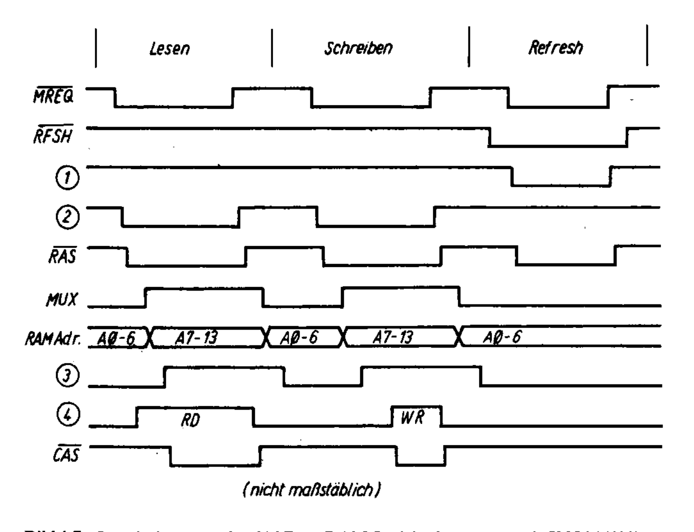


Bild 4.7 Impulsdiagramm der 64-kByte-RAM-Speicherbaugruppe mit *U 256 (4116)*

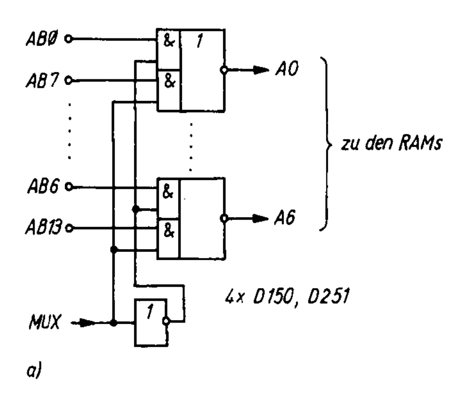
4.3. Aufbau und Inbetriebnahme

Die Inbetriebnahme der PROM-Platine und des statischen RAM-Speichers ist bei sauberem, übersichtlichem Aufbau problemlos.

Für den Einsatz des EPROM *U 555 (2708)* sowie des dynamischen RAM *U 256 (4116)* ist jedoch folgendes zu beachten:
Um ein Zerstören der genannten Speichertypen auszuschließen, muß die Betriebsspannung $U_{BB} = -5$ *V beim Einschalten zuerst und beim Ausschalten zuletzt am Schaltkreis anliegen (Einzelheiten dazu im Abschnitt 10.1.).*
Die weiteren Hinweise zu Aufbau und Inbetriebnahme beschränken sich auf den dynamischen RAM-Speicher. Wie bereits oben erwähnt, ist der mittlere Leistungsverbrauch eines dynamischen 16-kBit-RAM nur gering: im aktiven Zustand etwa 420 mW, im Standby-Zustand maximal 20 mW. Die mittleren Betriebsströme betragen: $I_{DD} = 35$ mA, $I_{CC} = 10\ \mu$A, $I_{BB} = 200\ \mu$A. Wenn der RAM jedoch angesteuert wird, entstehen Spitzenströme bis zu 100 mA je RAM. Deshalb ist jeder Schaltkreis besonders an den Anschlüssen U_{DD} und U_{BB} mit Stützkondensatoren (je 100 nF) abzublocken. Die Leitungen für die Betriebsspannung und die Masse müssen ausreichend dimensioniert werden.
Die DIL-Schalter (oder Drahtbrücken) S1 . . . S4 ermöglichen das Ein- und Ausschalten einzelner 16-k-Blöcke. Man sollte zunächst nur einen 16-k-Block mit Speicherschaltkreisen bestücken. Wenn sich die Speicherzellen nicht beschreiben lassen oder gar ihre Information verändern, so werden mit einem Oszilloskop die Impulsbilder überprüft. Vor allen Dingen bei unterschiedlich schnellen Speichern ist ein wenig Fingerspitzengefühl beim Einstellen der Verzögerungszeiten notwendig. Die *RC*-Werte sind erprobt für folgende RAM-Typen:
MK 4116–3, D 416C–2 und *TMM 416P–3.*
War das Schreiben und Lesen erfolgreich, dann sollte man einen RAM-Test mit dem in Tabelle 4.2. dargestellten Programm (nach

Tabelle 4.2. HEXA-Tabelle RAM-Testprogramm

Adresse HEXA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FC00	06	00	21	00	**ST**	7D	AC	A8	77	23	7C	FE	**SP**	C2	05	FC
FC10	21	00	**ST**	7D	AC	A8	BE	C4	25	FC	23	7C	FE	**SP**	C2	13
FC20	FC	04	C3	02	FC	C5	0E	05	CD	21	F0	52	41	4D	20	46
FC30	65	68	6C	65	72	20	41	64	72	2E	00	0E	04	CD	21	F0
FC40	0E	05	CD	21	F0	1E	00	C1	C9	ED	FF	52	41	4D	54	45
FC50	53	54	00	C3	00	FC										



a)

b)

Bild 4.8 Adreßmultiplexer; a – mit *D 150*, b – mit *8212*

MOSTEK) anschließen. Das Programm arbeitet mit dem in Abschnitt 9. beschriebenen Monitor zusammen. Einzutippen ist das Programm ab Adresse FC00H. ST bedeutet H-Teil der Anfangsadresse des zu testenden Speicherbereichs, SP bedeutet H-Teil der Endadresse. Die 4 Byte muß man selbst eintragen.
Gestartet wird das Testprogramm ab Adresse FC00H oder durch Eingabe des Befehls RAMTEST. Der Test eines 16-kByte-Speichers dauert etwa 4 Minuten. Der Speicher läßt sich beliebig lange testen (Abbruch durch $\overline{RESET}$ oder $\overline{NMI}$). Alle fehlerhaften Speicheradressen werden ausgegeben. Auch wenn dieser Test erfolgreich war, ist noch nicht hundertprozentig sicher, daß der Speicher funktioniert. Es kann sein, daß die Zugriffszeit der RAM-Karte während des $\overline{M1}$-Zyklus nicht klein genug ist. Dann müssen die Zeitkonstanten der $\overline{RC}$-Glieder der Verzögerungskette geringfügig verkleinert werden.
Ein Hinweis noch zu den verwendeten Bauelementen: Als Multiplexer wurden Schaltkreise vom Typ *74157* eingesetzt.
Bild 4.8 zeigt 2 Alternativen. Um den Datenbus nicht zu belasten, benutzt man Treiberschaltkreise vom Typ *8216*.

5. Fernsehinterface

Die Kommunikation zwischen Mensch und Computer setzt geeignete Ein-/Ausgabemedien voraus. Von besonderer Bedeutung in diesem Zusammenhang sind Geräte bzw. Systemkomponenten, die den visuell unterstützten Dialog ermöglichen. Aus der Vielzahl der Displayvarianten hat sich als universelles Anzeigemedium für Ziffern- und Zeichendarstellung bis hin zur Graphik der Bildschirm durchgesetzt. Das gilt in besonderem Maße für den Bereich der Mikrocomputer. Die Industrie bietet hierfür spezielle Bildschirmgeräte (vom Schwarzweiß- bzw. einfarbigen Monitor bis hin zum Farbmonitor) an.
Für viele Belange, vor allem natürlich aus der Sicht des Amateurs, erweist sich für den genannten Zweck der Einsatz eines handelsüblichen Fernsehgeräts als technisch und ökonomisch sinnvoll.
Im folgenden wird ein relativ komfortables, problemlos nachbaubares Fernsehinterface beschrieben.

5.1. Zeichendarstellung auf dem Bildschirm

Das vom Fernsehen bekannte Darstellungsprinzip beruht auf der Helligkeitssteuerung des Elektronenstrahls in der Bildröhre. Der abhängig von der Videoinformation getastete Strahl überstreicht dabei in einem Zeilenraster den Bildschirm von oben nach unten. Das geschieht nach CCIR-Norm mit einer Zeilenfrequenz von 15625 Hz (d. h. einer Zeilendauer von 64 µs) und 625 Zeilen je Bild. Daraus ergibt sich eine Bildwechselfrequenz von 25 Hz. Durch den Trick des Zeilensprungverfahrens erzeugt man eine Rasterwechselfrequenz von 50 Hz. Durch das Verkoppeln der Bildfrequenz wird ein flimmerfreies Bild abgesichert.
Zum Darstellen der Zeichen auf dem Bildschirm sollte man auf das Zeilensprungverfahren verzichten und statt dessen 2 identische Halbbilder mit etwa der Hälfte der 625 Fernsehzeilen erzeugen. Damit ist die für eine ausreichende Flimmerfreiheit des Bildes notwendige Bildfrequenz von etwa 50 Hz garantiert.
Die Anzahl der Fernsehzeilen (gewählt wurden 320 je Bild) ermög-

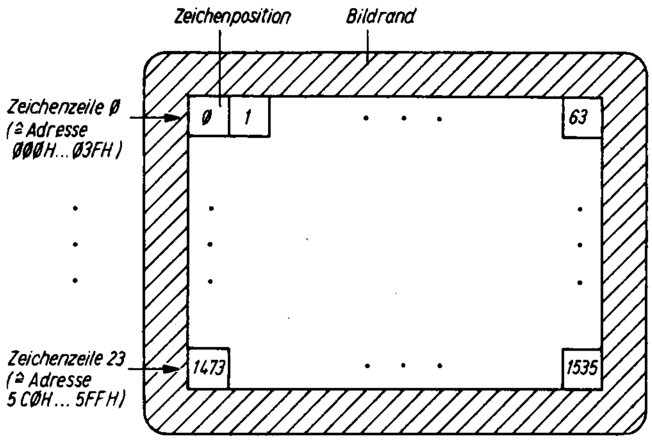


Bild 5.1 Bildschirmformat

licht die Darstellung einer relativ großen Anzahl von Zeichen auf dem Bildschirm (einschließlich des oberen und des unteren Bildrands).

Bild 5.1 zeigt das verwendete Bildschirmformat. Ein Bild besteht danach aus 24 Zeichenzeilen, jede Zeichenzeile aus 64 Zeichenpositionen. Damit ist die Darstellung von maximal 1536 Zeichen (d. h. 1,5 k) je Bild möglich.

Mit dieser Lösung wurde ein sinnvoller Kompromiß zwischen den technischen Möglichkeiten eines handelsüblichen TV-Geräts und der Forderung nach einer Abbildungsmöglichkeit für eine ausreichend große Zeichenzahl in gut lesbarer Form gefunden. Dabei nimmt man bewußt in Kauf, daß 0,5 kByte des 2-kByte-Bildwiederholspeichers nicht ausgenutzt sind.

Jeder Zeichenposition auf dem Bildschirm ist eine Adresse im Bildwiederholspeicher in steigender Reihenfolge fest zugeordnet. Der Bildschirmposition Ø entspricht also die Anfangsadresse CRTBG, der Position 1535 die Adresse CRTBG + 1535 des Bildwiederholspeichers.

Ein Zeichen wird auf dem Bildschirm nach dem häufig benutzten Prinzip der 5 × 7-Punktmatrix aufgebaut. Bild 5.2 zeigt die Zeichenzusammensetzung für das verwendete Bildschirminterface auf der Grundlage dieses Prinzips. Man benötigt zum Darstellen der

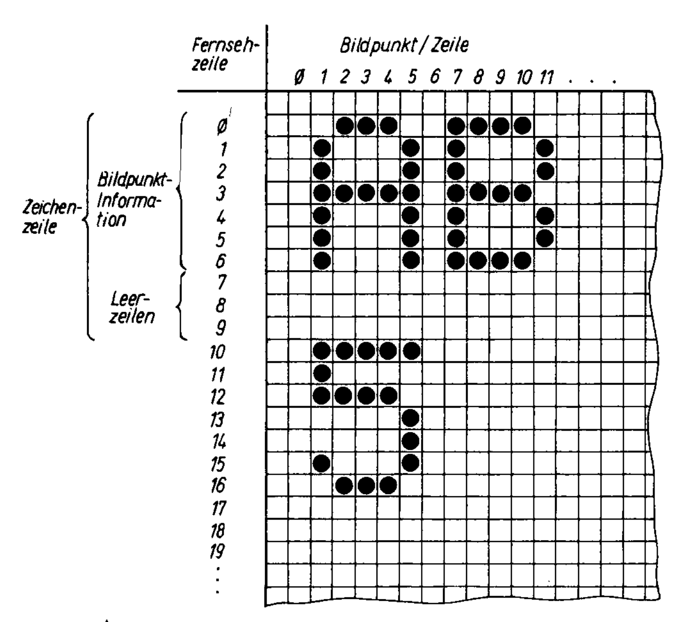


Bild 5.2 Zeichenaufbau nach dem Prinzip der 5 × 7-Punktmatrix

Information einer Zeichenzeile also 7 Fernsehzeilen. Zur Gewährleistung eines übersichtlichen, gut lesbaren Bildes wird jede Zeichenzeile durch 3 weitere Fernsehzeilen von der benachbarten getrennt. Insgesamt besteht damit eine Zeichenzeile aus 10 Fernsehzeilen.

Für jedes Einzelzeichen stehen in der Horizontalen, d. h. je Fernsehzeile, 6 Bildpunkte zur Verfügung. Jeweils der 1. oder 6. Bildpunkt (abhängig von der gewählten Bitposition auf dem Zeichengenerator) wird dunkelgetastet und bildet den horizontalen Zeichenabstand.

Unter Berücksichtigung von Bildschirmformat und Zeichenaufbau setzt sich das Bildfeld damit aus 240 Fernsehzeilen zusammen. Eine Zeile besteht aus 384 Bildpunkten.

Bild 5.3 zeigt das BAS-Signal (Bild-, Austast- und Synchronsignal) nach CCIR, das mit dem Bildschirminterface erzeugt werden muß, um ein handelsübliches TV-Gerät ansteuern zu können. Danach

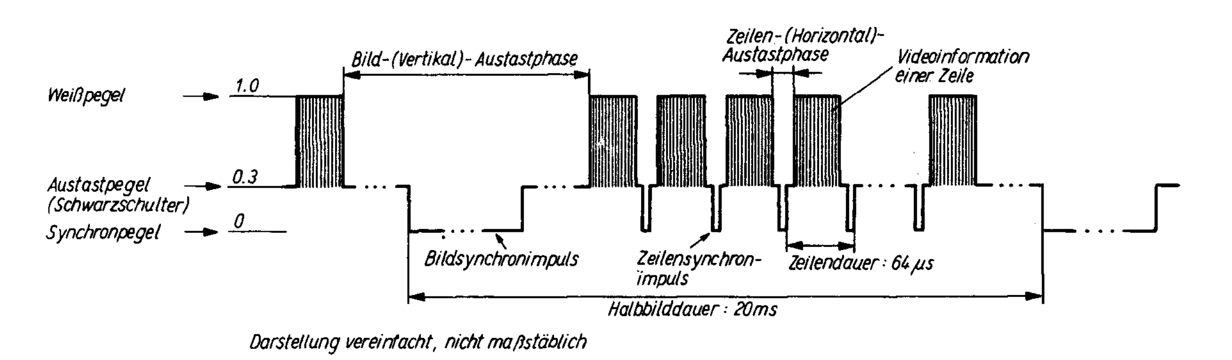


Bild 5.3 BAS-Signal nach CCIR

ist neben der Videoinformation, die die 384 Bildpunkte je Zeile abhängig von dem jeweils darzustellenden Zeichen hell- und dunkeltastet, auch die Bereitstellung der Zeilen- und Bildsynchronsignale sowie der entsprechenden Austastsignale notwendig.
Durch das Zeilenaustasten entstehen ein linker und ein rechter Bildrand. In die Zeilenaustastphase fällt auch der Strahlrücklauf, der damit nicht sichtbar ist. Bei der Bildaustastung entstehen ein oberer und ein unterer Bildrand.
Durch die Festlegung von Bildrändern wird abgesichert, daß alle 24 Zeichenzeilen zu je 64 Zeichen eindeutig erkennbar sind. Es werden also keine Zeichen von den Bildschirmrändern „verschluckt". Die an den Bildschirmrändern häufig auftretenden Unschärfen und Verzeichnungen haben keine Auswirkungen.

5.2. Schaltungsbeschreibung

Bild 5.4 zeigt einen Übersichtsschaltplan, an dem das grundsätzliche Funktionsprinzip des Fernsehinterface des Mikrocomputers erklärt werden soll.
Der Zugriff der CPU zum Bildschirm ist über den Bildwiederholspeicher sichergestellt, da dieser wie ein normaler RAM-Speicher (siehe Abschnitt 4.2.) verwaltet wird. In diesem Fall sind die Adreßleitungen des Bildwiederholspeichers über den Adreßmultiplexer auf den Systembus geschaltet.
Die CPU kann jetzt mit einem normalen Speicherschreibzyklus den Zeichencode (ISO-7-Bit bzw. ASCII) auf der gerade gewählten Adresse im Bildwiederholspeicher ablegen. Damit ist, wie bereits in Abschnitt 5.1. erwähnt, auch die Zeichenposition auf dem Bildschirm festgelegt.
In dieser Art wird der gesamte auf dem Bildschirm darzustellende Bildinhalt im Bildwiederholspeicher abgespeichert.
Nachdem der Zugriff der CPU auf den Bildwiederholspeicher beendet ist, schaltet der Adreßmultiplexer um, und die umfangreiche Elektronik der Interfaceschaltung sorgt dafür, daß der vollständige Inhalt des Bildwiederholspeichers in eine zum Tasten des Elektronenstrahls geeignete Videoinformation umgewandelt und zyklisch mit der Bildfrequenz wiederholt wird. Ein Taktgenerator liefert den Bildpunkttakt und steuert damit eine Zählerkette an. Diese Zählerkette wurde so gestaltet, daß die entsprechenden Zähler-

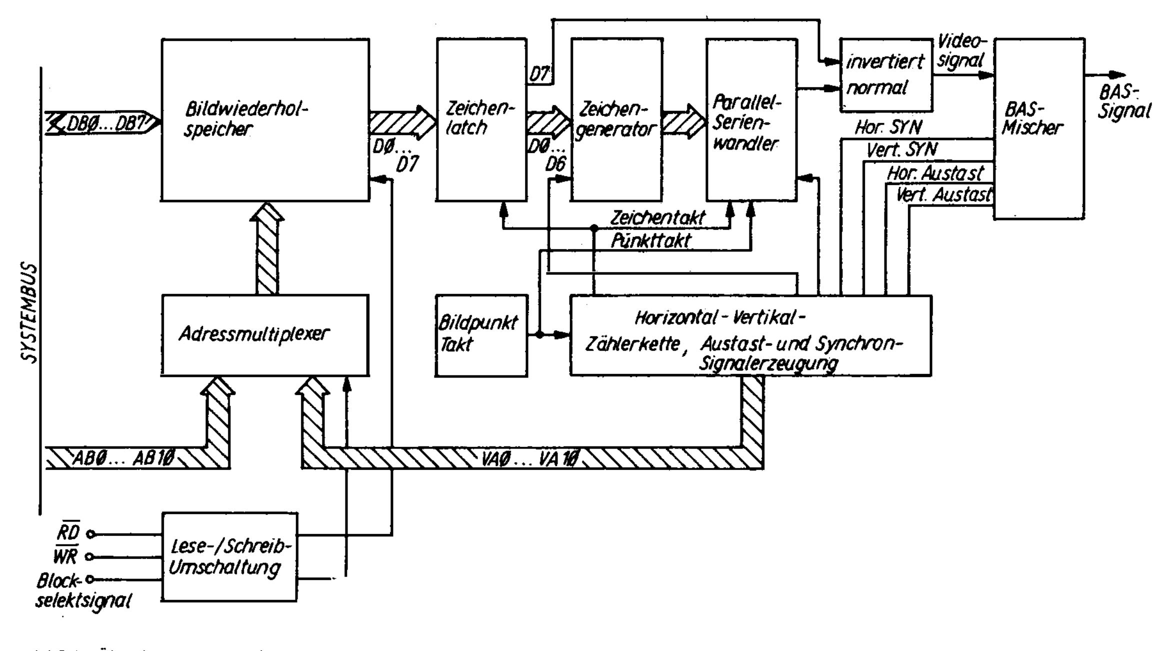
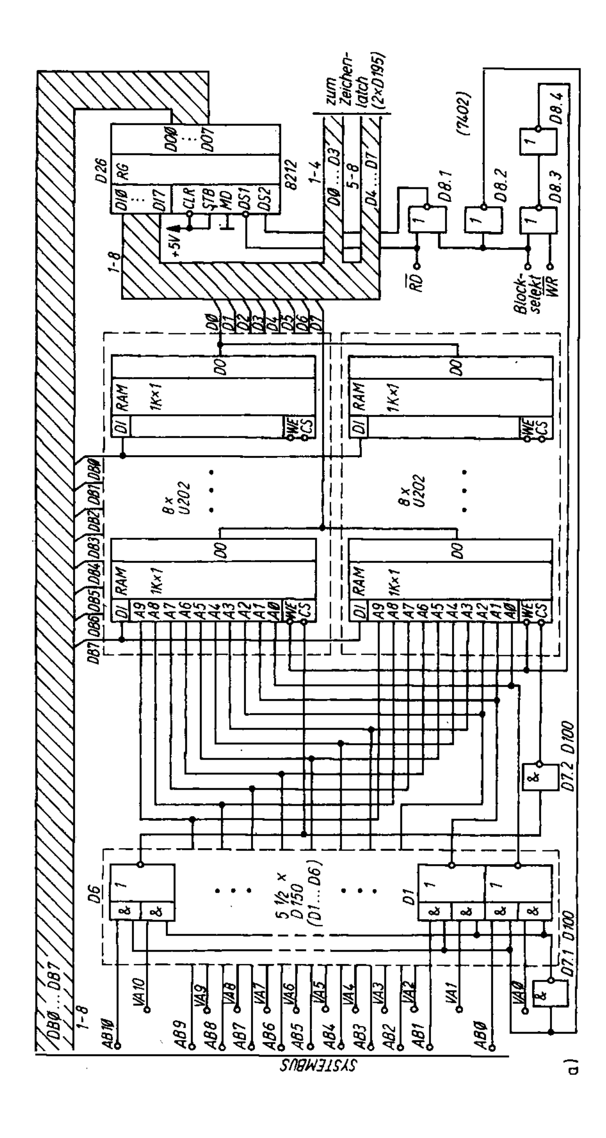


Bild 5.4 Übersichtsschaltplan des Fernsehinterface

ausgänge über den Adreßmultiplexer direkt zur zyklischen Adressierung des Bildwiederholspeichers herangezogen werden können. Diese Adreßinformation wird synchron mit der dazugehörigen Position des Elektronenstrahls auf dem Bildschirm bereitgestellt.
Aus den Zählerzuständen gewinnt man außerdem durch geeignete logische Verknüpfung die Zeilen- und Bildsynchronimpulse sowie die dazugehörigen Austastsignale.
Die gerade adressierte Speicherzelle im Bildwiederholspeicher stellt den aktuellen Zeichencode über den Zeichenlatch dem Zeichengenerator zur Verfügung. Im Zeichengenerator, einem Festwertspeicher (PROM), sind die der jeweiligen Zeichenkonfiguration (siehe Bild 5.2) entsprechenden Bildpunktinformationen abgespeichert und liegen an den Ausgängen parallel in 8-Bit-Breite vor. Am Ausgang des anschließenden Parallel-Serienwandlers läßt sich jetzt die serielle Videoinformation abnehmen. Sie wird danach mit dem zur Zeichenauswahl (maximal 128 verschiedene Zeichen) nicht benötigten Bit 8 des Bildwiederholspeichers exclusiv – oder verknüpft. Damit ist die Zusatzfunktion „Zeicheninvertierung“ möglich.
Synchronaustast- und Videosignale werden am BAS-Mischer zusammengeführt und zu dem in Bild 5.3 dargestellten BAS-Signal nach CCIR-Norm verarbeitet.

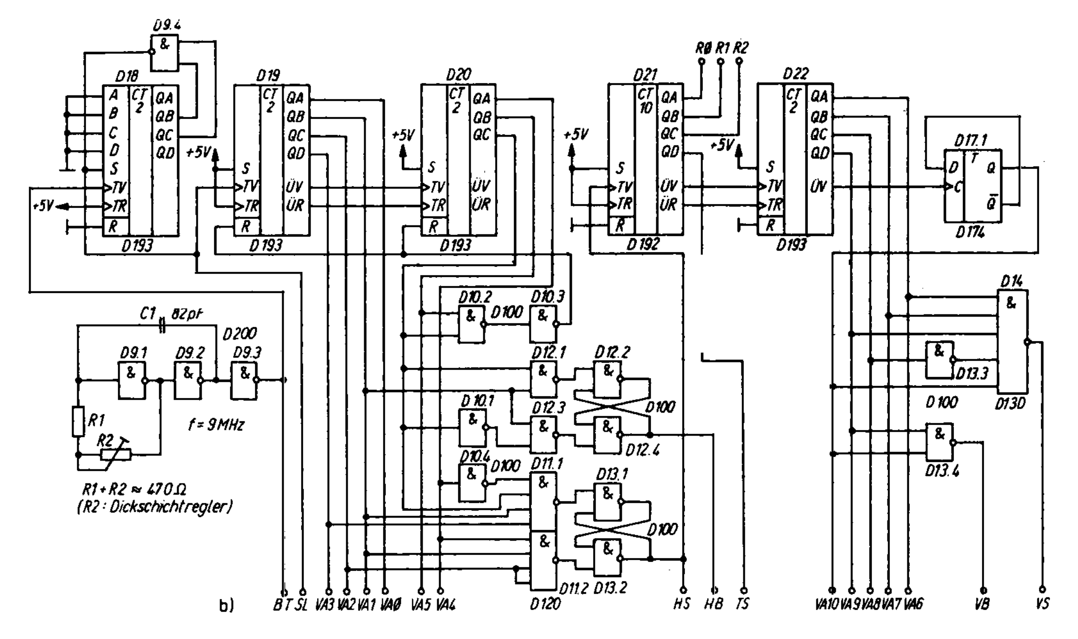
5.2.1. Bildwiederholspeicher, Adreßmultiplexer

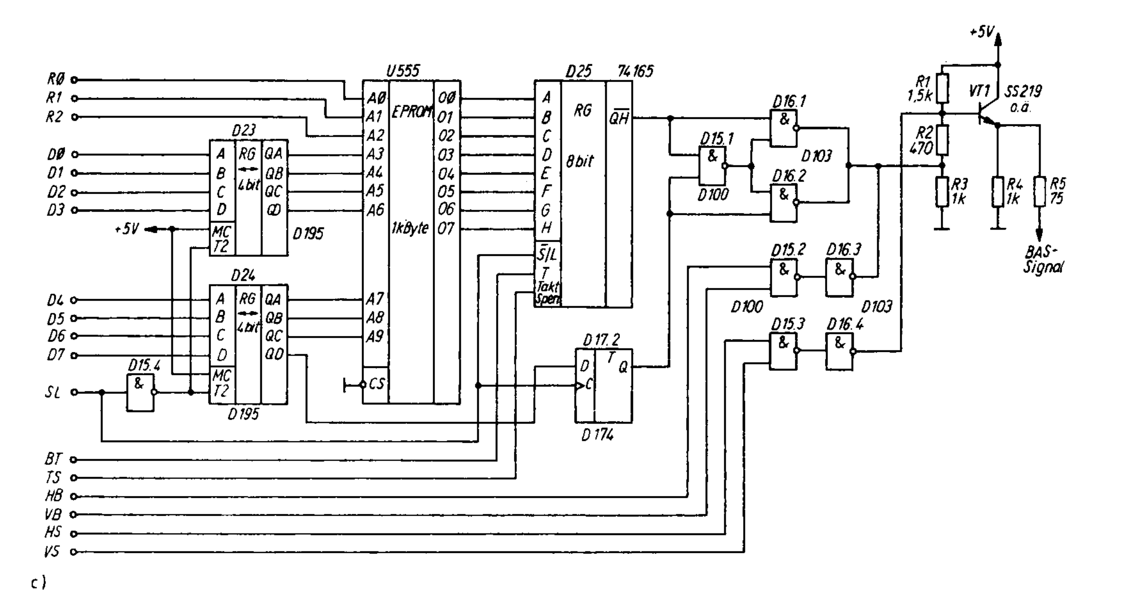
Bild 5.5 zeigt die gewählte Schaltung für das Bildschirminterface. Den Bildwiederholspeicher muß man so konzipieren, daß in ihm der gesamte Bildinhalt gespeichert werden kann. Im vorliegenden Beispiel sind das die Codes für 1536 Zeichen, d. h. 1,5 kByte.
Der Bildwiederholspeicher ist wieder mit statischen RAM *U 202* aufgebaut. Auf Grund seiner 1024 × 1-Bit-Organisation sind also in der bereits bekannten Weise 2 RAM-Blöcke zu je 1 kByte Speicherkapazität zu realisieren. Damit bleiben (wie bereits erläutert wurde) 0,5 kByte ungenutzt. Prinzipiell ließe sich dieser Rest natürlich als normaler RAM-Speicher nutzen. Zu empfehlen ist dies aber nicht, da sich (ohne besondere schaltungstechnische Maßnahmen) jeder CPU-Zugriff zum Bildwiederholspeicher durch einen kurzzeitigen undefinierten Bildschirmzustand bemerkbar macht. Der würde natürlich auch beim Ansprechen des 0,5-kByte-Rests auftreten, obwohl sich am Bildinhalt selbst nichts änderte.



zum Zeichen-latch (2×D195)
D26
8212
(7402)
RD
Block-selekt
WR
8 x U202
RAM
1K×1
5 1/2 x D150 (D1...D6)
D7.1 D100
D7.2 D100
SYSTEMBUS

a)


D9.4
D18
D19
D20
D21
D22
D193
D192
D17.1
D174
D14
D13.3
D13D
D13.4
D100
D10.2
D10.3
D12.1
D12.2
D10.1
D12.3
D12.4
D10.4
D11.1
D13.1
D13.2
D11.2
D120
D9.1
D9.2
D9.3
D200
C1 82pF
f = 9MHz
R1
R2
R1 + R2 ≈ 470 Ω
(R2: Dickschichtregler)
+5V
RØ R1 R2
BT SL VA3 VA2 VA1 VAØ VA5 VA4
HS HB TS
VA10 VA9 VA8 VA7 VA6
VB VS
b)



c)

Bild 5.5 a – Bildwiederholspeicher, Adreßmultiplexer, b – Synchron- und Austastsignalerzeugung, Adressenbereitstellung, c – Steuerteil

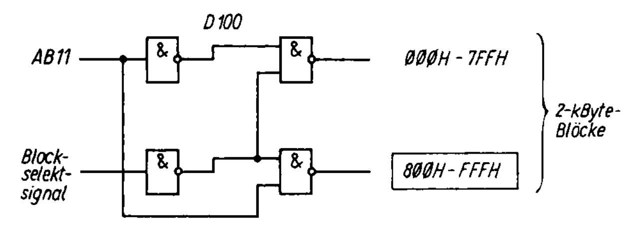


Bild 5.6 Selektierung des Bildwiederholspeichers

In Abhängigkeit davon, ob die CPU Zugriff zum Bildschirminterface hat oder nicht, müssen die Adreßleitungen A0 ... A10 des Bildwiederholspeichers entweder auf den Datenbus des Systems oder auf die zur zyklischen Adressierung notwendigen Zählerausgänge gelegt werden. Dieses Umschalten bewirkt ein Adreßmultiplexer, der mit Schaltkreisen des Typs *D 150* aufgebaut wurde (D1 ... D6). Eleganter wäre der Einsatz von 3 Schaltkreisen vom Typ *74157*. Dieser Schaltkreis beinhaltet jeweils vier 2-auf-1-Multiplexer [3].
Über die NOR-Gatter von D6 wird der Bildwiederholspeicher bei E8ØØH (siehe Bild 3.3) selektiert und der Adreßmultiplexer umgeschaltet.
An Stelle des Treiberschaltkreises *8212* eignen sich selbstverständlich auch 2 Exemplare des Typs *8216*.
Das 4-kByte-Blockselektsignal (EØØØH) wird wieder der zentralen Speicherselektierung entnommen, wobei man entsprechend der Speicherkapazität des Bildwiederholspeichers und der gewählten Adresse E8ØØH noch in zwei 2-kByte-Blöcke unterteilen muß. Dies ist durch 1-aus-2-Decodierung des Blockselektsignals und des Adreßbit AB11 möglich (Bild 5.6).

5.2.2. Synchron- und Austastsignalerzeugung, Adressenbereitstellung

Den Bildpunkttakt liefert ein freischwingender Impulsgenerator, bestehend aus den schnellen Gattern des Typs *D 200* (D 9.1 ... D 9.3). Als Taktfrequenz wurde f_{BP} = 9 MHz ausgewählt. Damit ergeben sich günstige Zeitverhältnisse.
Bildpunktdauer: t_{BP} = 111 ns; mit 64 darstellbaren Zeichen je Zei-

chenzeile und 6 Bildpunkten je Zeichen und Fernsehzeile erhält man 384 Bildpunkte je Fernsehzeile. Daraus folgt eine genutzte Zeilendauer von 384 · 111 ns = 42,6 μs. Das bedeutet also, daß 42,6 μs der verfügbaren Zeilendauer von 64 μs für das Schreiben einer Zeile des Bildfelds genutzt werden. Der verbleibende Rest von 21,4 μs wird dunkel getastet und ergibt den rechten und den linken Bildrand.

Mit dem Bildpunkttakt wird ein schneller 6 : 1-Zähler auf der Basis des 4-Bit-Binärzählers *D 193* (D18) angesteuert. Er zählt die 6 Bildpunkte, die je Fernsehzeile zum Darstellen eines Zeichens notwendig sind, und gibt danach den Impuls an den folgenden Zeichenpositionszähler ab. Dieser besteht aus 2 Schaltkreisen *D 193* (D19, D20) und zählt die Zeichenpositionen je Zeichenzeile bis 64. Die Gatter D10.2 und D10.3 übernehmen das Rücksetzen dieses Zählers nach 96 Eingangsimpulsen (d. h. 576 Bildpunktimpulsen).

Aus den Zählerzuständen des Zeichenpositionszählers werden in geeigneter Form der Zeilensynchronimpuls (Gatter D10.4, D11, D13.1, D13.2) und der Zeilenaustastimpuls (Gatter D12, D10.1) decodiert. Für den Zeilensynchronimpuls ergeben sich eine Impulsbreite von t_{IH} = 8 μs und die geforderte Periode von T_H = 64 μs. Die Austastzeit beträgt, wie bereits gezeigt, t_{AH} = 21,4 μs. Bild 5.7 zeigt die zeitlichen Abläufe.

Man erkennt die um 2 Zeichenpositionen (12 Bildpunkte entsprechend 1,33 μs) verzögerte Helltastung. Damit werden keine undefinierten Zeichen auf dem Bildschirm sichtbar.

Der Ausgangsimpuls der bisher beschriebenen Horizontalzählerkette, d. h. der Zeilensynchronimpuls, gelangt auf den Eingang der sich anschließenden Vertikalzählerkette. Der am Beginn dieser Zählerkette befindliche Dezimalzähler *D 192* (D21) erhält also mit jeder Fernsehzeile, die der Elektronenstrahl auf den Bildschirm geschrieben hat, einen Ansteuerimpuls. Er zählt die zur Darstellung einer Zeichenzeile notwendigen 10 Fernsehzeilen (7 für die eigentliche Zeicheninformation, 3 für den vertikalen Zeichenzeilenabstand).

Nach Beendigung dieses Vorgangs wird ein Impuls an den nachfolgenden Zeichenzeilenzähler abgegeben. Dieser besteht aus dem Schaltkreis *D 193* (D22) sowie einem Binäruntersetzer *D 174* (D17.1) und zählt bis 32. Da nach 32 Eingangsimpulsen der Zählvorgang wieder bei 0 beginnt, ist kein Rücksetzen erforderlich.

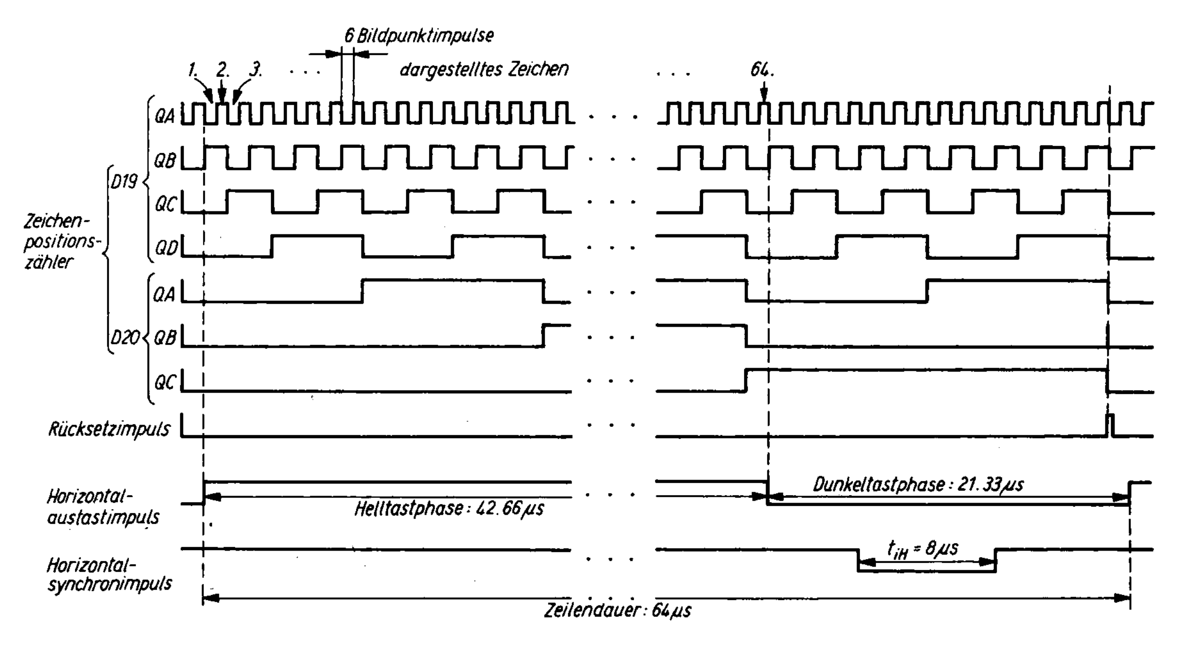


Bild 5.7 Impulsdiagramm der Horizontalablenkung

Das gesamte Bild besteht damit aus 32 Zeichenzeilen, entsprechend also 320 Fernsehzeilen. Davon entfallen 24 Zeichenzeilen auf das Bildfeld, die verbleibenden 8 (d. h. 80 Fernsehzeilen) bilden den oberen und den unteren Bildrand. Es ergibt sich daraus eine Bildfrequenz von

$$\frac{1}{320 \cdot 64\,\mu s} = 48{,}8\ \text{Hz.}$$

Damit ist eine ausreichende Flimmerfreiheit des Bildes gewährleistet.
Aus den Zählerzuständen des Zeichenzeilenzählers werden mit der Gatterkombination D13.3, D14 der Vertikalsynchronimpuls ($t_{IV} = 640\,\mu s$) sowie der Vertikalaustastimpuls ($t_{AV} = 80 \cdot 64\,\mu s = 5{,}12$ ms) mit dem Gatter D13.4 decodiert.
Bild 5.8 zeigt die Zeitverhältnisse für die Vertikalimpulserzeugung.
Über die zyklische Adressierung durch die entsprechenden Zählerausgänge werden dem Bildwiederholspeicher die *Zeichenposition (0 ... 63)* und die *Zeichenzeile (0 ... 23)* signalisiert, in der sich der Elektronenstrahl gerade befindet. Genau aus diesem Sachverhalt ergibt sich die eindeutige Zuordnung zwischen Zeichenpositionen auf dem Bildschirm und Adresse im Bildwiederholspeicher.
Die Ausgänge QA, QB, QC des Dezimalzählers *D 192* sind mit den Adreßeingängen A0, A1, A2 des im weiteren noch zu beschreibenden Zeichengenerators verbunden. Damit wird dem Zeichengenerator mitgeteilt, in welcher Fernsehzeile des gerade darzustellenden Zeichens sich der Elektronenstrahl befindet. Der Ausgang QD des *D 192* sorgt über den Taktsperreingang des Parallel-Serienwandlers *74165* dafür, daß in den letzten 2 Fernsehzeilen (d. h. während der Zählerzustände 8, 9) jeder Zeichenzeile keine Videoinformation ausgegeben wird. Diese beiden sowie eine weitere, auf dem Zeichengenerator erzeugte, Leerzeile trennen die Zeichenzeilen in der Vertikalen.

5.2.3. Videosignalerzeugung

Der aktuelle Zeichencode, d. h. der Inhalt der gerade im Bildwiederholspeicher adressierten Speicherzelle, wird mit jedem Ausgangsimpuls des Bildpunktzählers in den Zeichenlatch übernommen und steht gleichzeitig an dessen Ausgang zur Verfügung.

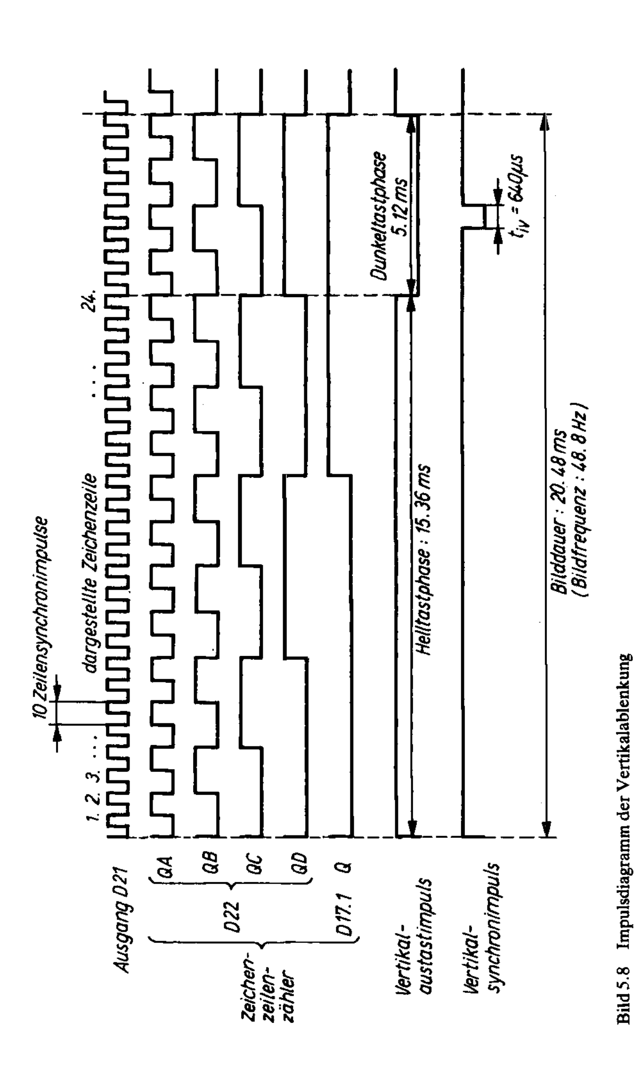


Bild 5.8 Impulsdiagramm der Vertikalablenkung

Der Zeichenlatch wurde mit zwei 4-Bit-Schieberegistern des Typs *D 195* aufgebaut. Er stellt dem Zeichengenerator unter Berücksichtigung der Speicherzugriffszeiten den Zeichencode als stabile Adresse zur Verfügung.

Der Zeichencode bildet also über die Adreßeingänge A3 ... A9 eine von 128 möglichen Adressen für den Zeichengenerator-PROM. Auf genau dieser Adresse sind die Bildpunktinformationen für die 7 Zeilen des darzustellenden Zeichens in 7 aufeinanderfolgenden Speicherzellen als Datenbyte abgespeichert. Die 8. Speicherzelle in dieser Reihenfolge ist mit 00H belegt und bildet damit die 1. der 3 Leerzeilen für den vertikalen Zeichenzeilenabstand.
Es werden nur 6 Bit der 8-Bit-Speicherbreite des PROM genutzt (5 Bit Bildpunktinformation, 1 Bit horizontaler Zeichenabstand). Insgesamt benötigt man damit zum Darstellen eines Zeichens 8 aufeinanderfolgende Speicherzellen des Zeichengenerator-PROM. Die 8 Speicherzellen werden, wie in Abschnitt 5.2.2. bereits erwähnt, durch die Zählerausgänge QA, QB, QC des Dezimalzählers *D 192* an den Adreßeingängen A0, A1, A2 des Zeichengenerators adressiert.
Tabelle 5.1. enthält die Bildpunktinformation für den Buchstaben

Tabelle 5.1. Zuordung: Zeichencode – Speicherzelle – Bildpunktinformation auf dem Zeichengenerator-PROM

Zeichencode (A3 ... A9)	PROM-Speicherzelle (A0 ... A9)	Zeichen D7 ... D0	Datenbytes	
41H	208 H	· · ● ● ● · · ·	38H	Bildpunktinformation
	209 H	· ● · · · ● · ·	44H	
	20A H	· ● · · · ● · ·	44H	
	20B H	· ● ● ● ● ● · ·	7CH	
	20C H	· ● · · · ● · ·	44H	
	20D H	· ● · · · ● · ·	44H	
	20E H	· ● · · · ● · ·	44H	
	20F H	· · · · · · · ·	00H	Leerzeile

D7: horizontaler Zeichenabstand; D1, D0: nicht belegt

Beispiel: Adressbitbelegung Zeichengenerator-PROM

	A9 ... A3 ... A0
	0 0 1 0 0 0 0 0 1 0 0 0
Speicherzelle: (HEXA)	2 0 8
Zeichencode: (HEXA)	4 1

Tabelle 5.2. HEXA-Tabelle Zeichengenerator

0000H/03FFH

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	00	08	18	38	78	38	18	08	00	28	28	28	28	28	28	28	0010	00	7C	00	44	28	10	28	44	00	00	00	7E	00	7E	00	00
0020	00	7C	00	58	64	44	44	44	00	00	00	34	48	48	48	34	0030	00	30	48	48	58	44	58	40	00	18	20	10	38	44	44	38
0040	00	00	00	00	10	28	44	7C	00	00	40	70	48	48	08	04	0050	00	38	44	44	7C	44	44	38	00	00	40	20	10	28	44	44
0060	00	00	00	24	24	24	38	40	00	00	04	38	68	28	28	28	0070	00	00	00	3C	48	48	48	30	00	7C	20	10	08	10	20	7C
0080	00	00	00	04	38	50	10	10	00	00	10	38	54	54	38	10	0090	00	38	44	44	44	44	28	60	00	10	00	38	44	7C	44	44
00A0	00	10	00	38	48	48	48	34	00	28	00	38	44	7C	44	44	00B0	00	28	00	38	48	48	48	34	00	28	38	44	44	44	44	38
00C0	00	28	00	38	44	44	44	38	00	28	00	44	44	44	44	38	00D0	00	00	28	00	44	44	44	38	00	00	10	08	7C	08	10	00
00E0	00	00	3C	20	20	20	60	20	00	30	48	10	20	78	00	00	00F0	00	18	24	20	70	20	20	7C	00	44	28	10	10	38	10	10
0010	00	00	00	00	00	00	00	00	00	20	20	20	20	20	00	20	0110	00	28	28	00	00	00	00	00	00	28	28	7C	28	7C	28	28
0120	00	10	3C	50	38	14	78	10	00	60	64	08	10	20	4C	0C	0130	00	20	50	20	54	48	34	00	30	30	10	10	20	00	00	00
0140	00	08	10	10	10	10	10	08	00	20	10	10	10	10	10	20	0150	00	00	10	54	38	54	10	00	00	00	10	10	7C	10	10	00
0160	00	00	00	00	30	30	10	20	00	00	00	00	7C	00	00	00	0170	00	00	00	00	00	30	30	00	00	00	04	08	10	20	40	00
0180	00	38	44	4C	54	64	44	38	00	10	30	10	10	10	10	38	0190	00	38	44	04	18	20	40	7C	00	38	44	04	18	04	44	38
01A0	00	08	18	28	48	7C	08	08	00	7C	40	78	04	04	44	38	01B0	00	18	20	40	78	44	44	38	00	7C	04	08	10	20	20	20
01C0	00	38	44	44	38	44	44	38	00	38	44	44	3C	04	08	30	01D0	00	00	30	30	00	30	30	00	00	30	30	00	30	30	10	20
01E0	00	04	08	10	20	10	08	04	00	00	00	7C	00	7C	00	00	01F0	00	40	20	10	08	10	20	40	00	38	44	04	08	10	00	10
0200	00	38	44	5C	54	5C	40	38	00	38	44	44	7C	44	44	44	0210	00	78	44	44	78	44	44	78	00	38	44	40	40	40	44	38
0220	00	78	44	44	44	44	44	78	00	7C	40	40	78	40	40	7C	0230	00	7C	40	40	78	40	40	40	00	38	44	40	40	4C	44	3C
0240	00	44	44	44	7C	44	44	44	00	38	10	10	10	10	10	38	0250	00	04	04	04	04	04	44	38	00	44	48	50	60	50	48	44
0260	00	40	40	40	40	40	40	7C	00	44	6C	54	54	44	44	44	0270	00	44	44	64	54	4C	44	44	00	38	44	44	44	44	44	38
0280	00	78	44	44	78	40	40	40	00	38	44	44	44	54	48	34	0290	00	78	44	44	78	50	48	44	00	38	44	40	38	04	44	38
02A0	00	7C	10	10	10	10	10	10	00	44	44	44	44	44	44	38	02B0	00	44	44	44	28	28	10	10	00	44	44	44	44	54	54	6C

Tabelle 5.2. HEXA-Tabelle Zeichengenerator

0000H/03FFH

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F																	
02C0	00	44	44	28	10	28	44	44	00	44	44	28	10	10	10	10	02D0	00	7C	04	08	10	20	40	7C	00	1C	10	10	10	10	10	1C
02E0	00	00	40	20	10	08	04	02	00	70	10	10	10	10	10	70	02F0	00	10	38	54	10	10	10	10	00	00	00	00	00	00	00	7C
0300	00	30	30	20	10	00	00	00	00	00	00	38	48	48	48	34	0310	00	40	40	58	64	44	44	78	00	00	00	38	40	40	44	38
0320	00	04	04	34	4C	44	44	3C	00	00	00	38	44	78	40	38	0330	00	10	28	20	70	20	20	20	00	00	00	3C	44	3C	04	18
0340	00	40	40	58	64	44	44	44	00	10	00	30	10	10	10	38	0350	00	08	00	18	08	08	48	30	00	20	20	24	28	30	28	24
0360	00	30	10	10	10	10	10	38	00	00	00	28	54	54	44	44	0370	00	00	00	58	64	44	44	44	00	00	00	38	44	44	44	38
0380	00	00	00	78	44	64	58	40	00	00	00	38	48	58	28	0C	0390	00	00	00	2C	30	20	20	20	00	00	00	38	40	38	04	78
03A0	00	00	20	70	20	20	28	10	00	00	00	44	44	44	4C	34	03B0	00	00	00	44	44	44	28	10	00	00	00	44	44	54	54	28
03C0	00	00	00	44	28	10	28	44	00	00	00	44	28	10	10	20	03D0	00	00	00	7C	08	10	20	7C	00	08	10	10	20	10	10	08
03E0	00	10	10	10	00	10	10	10	00	20	10	10	08	10	10	20	03F0	00	00	00	20	54	08	00	00	00	28	54	28	54	28	54	28

A sowie die entsprechenden Datenbyte, die auf dem Zeichengenerator abzuspeichern sind. Auch die Zuordnung zwischen Speicherzellenadressierung (A0 ... A9) und Zeichencode (A3 ... A9) ist aus der Tabelle ersichtlich.
Als Zeichengenerator wird der 1-kByte-PROM *U 555* verwendet. Über dessen Adreßeingänge A3 ... A9 lassen sich also 128 Zeichen zu je 8 aufeinanderfolgenden Speicherzellen (das entspricht genau 1 kByte) auswählen.
Für die notwendigen Zeichencodevereinbarungen gilt der ISO-7-Bit- bzw. ASCII-Code (siehe Tabellen im Anhang). Die Auswahl der 128 Zeichen und das Programmieren des Zeichengenerators-PROM sind unter Berücksichtigung der gegebenen Hinweise (Zeichenaufbau, Adreßbelegung, Codevereinbarung) nach eigenem Geschmack möglich. Beispielsweise können selbst zu gestaltende Pseudographikelemente mit vorgesehen werden.
In Tabelle 5.2. ist ein bewährter Zeichengenerator in Form einer Hexatabelle angegeben. Er enthält neben den Ziffern, den großen und kleinen Buchstaben alle Umlaute, die gebräuchlichsten griechischen Buchstaben sowie alle aus der Sicht der Thematik relevanten Sonderzeichen.
Entsprechend der Adressierung durch den jeweiligen Zeichencode liegt also die aktuelle Bildpunktinformation einer Zeile des gerade darzustellenden Zeichens an den Ausgängen 00 ... 07 des Zeichengenerators *U 555* parallel vor. Der sich anschließende Parallel-Serienwandler wandelt diese Information in ein serielles Videosignal um, wie es zum Tasten des Elektronenstrahls benötigt wird.
Zur Parallel-Serienwandlung wird das 8-Bit-Schieberegister *74165* verwendet [3]. Für die Übernahme der Parallelinformation vom Zeichengenerator (d. h. zum Laden des Schieberegisters) ist ein Nadelimpuls erforderlich, der in dem Moment an den Shift-Load-Eingang des *74165* geliefert wird, wenn der Elektronenstrahl die 6 Bildpunkte des vorhergehenden Zeichens in der jeweiligen Fernsehzeile auf den Bildschirm geschrieben hat.
Der benötigte Ladeimpuls steht am Ausgang des Bildpunktzählers zur Verfügung. Er wird durch Gatter D15.4 invertiert und zum Laden des Zeichenlatch *D 195* verwendet.
Die Videoinformation muß selbstverständlich synchron mit dem Bildpunkttakt seriell ausgegeben werden. Zu dem Zweck ist der Takteingang des Schaltkreises *74165* direkt mit dem Ausgang des Taktgenerators verbunden.

Die sinnvolle Verwendung des Taktsperreingangs am Schieberegister *74165* wurde bereits beschrieben.
Im Abschnitt 5.2. wurde darauf hingewiesen, daß das Bit 8 des Bildwiederholspeichers zur Auswahl der 128 auf dem Zeichengenerator programmierten Zeichen nicht benötigt wird. Man realisiert deshalb damit die Zusatzfunktion „Zeicheninvertierung", indem das Bit 8 nach Zwischenspeicherung im D-Latch *D 174* (D17.2) mit dem Videosignal exclusiv – oder verknüpft (Gatter D15.1, D16.1, D16.2) wird. Für den Fall, daß Bit 8 H-Pegel führt, ist eine invertierte Zeichendarstellung (dunkel auf hellem Grund) sichergestellt.

5.2.4. BAS-Signalerzeugung

Ein TV-Gerät läßt sich nur mit BAS-Signal nach CCIR gemäß Bild 5.3 ansteuern. Dieses erzeugt man, indem das Videosignal mit den vorher beschriebenen Synchron- und Austastsignalen zusammengeführt wird. Hierzu werden die Horizontal- und Vertikalsynchronsignale am Gatter D15.3 und die entsprechenden Austastsignale am Gatter D15.2 und verknüpft. An den Ausgängen der Open-Collektor-Gatter *D 103* (D16) liegen jetzt alle Signale in richtiger Polarität vor. Der BAS-Mischer, Transistor VT 1, mischt sie und liefert am Ausgang polaritäts- und potentialgerecht das BAS-Signal. Das BAS-Signal wird dem Video-Eingang des TV-Geräts zugeführt.
Für bestimmte Fernsehgeräte (z. B. *Combivision*) hat es sich als günstig erwiesen, Video- und Synchronsignale getrennt den entsprechenden Stufen im TV-Gerät zuzuführen.
Verwendet man einen geeigneten HF-Modulator, kann das BAS-Signal auch direkt über den Antenneneingang eingespeist werden. Hierbei sind die Störstrahlungsbestimmungen der Deutschen Post zu beachten!

5.3. Erweiterungsmöglichkeiten

Das Fernsehinterface kann durch die mögliche Darstellung graphischer Informationen (z. B. in Verbindung mit den in Abschnitt 7. vorgestellten AD-Wandlern) auf dem Bildschirm sinnvoll erweitert werden. Dazu bieten sich mehrere Möglichkeiten.

5.3.1. Zeichengenerator mit Graphiksymbolen

Auf dem Zeichengenerator werden neben dem ASCII- bzw. ISO-7-Bit-Zeichensatz Graphikelemente (z. B. Blockgraphik) untergebracht. Diese Blockelemente gestaltet man so, daß sich aus ihnen die gewünschten graphischen Darstellungen auf dem Bildschirm zusammensetzen lassen. Als Zeichengenerator eignet sich ein PROM mit einer Speicherkapazität ≧ 2 kByte (z. B. *2716, 2732*). Dabei wird der Ausgang QD des Zählers *D 192* (D21) vom Taktsperreingang des Parallel-Serienwandlers *74165* (D25) abgetrennt und zum Adressieren des PROM herangezogen. Jetzt können alle 10 Fernsehzeilen einer Zeichenposition zur Informationsdarstellung genutzt werden (z. B. auch bei Buchstaben zum Erzeugen von Unterlängen).

5.3.2. RAM-Speicher als Zeichengenerator

Als Zeichengenerator wird ein RAM-Speicher verwendet. Die gewünschten Graphiksymbole (z. B. Kurvenabschnitt u. ä.) lassen sich jetzt nach Bedarf programmieren. Zu diesem Zweck werden die Datengänge der RAM (ggf. über Treiberschaltkreise) auf den Datenbus gelegt. Die Datenausgänge sind unmittelbar mit den Eingängen des Schieberegisters *74165* verbunden.
Die Adreßeingänge werden über 2-auf-1-Multiplexer bei CPU-Zugriff auf den Adreßbus und andernfalls auf die Latchausgänge *D 195* (D23, D24) bzw. die Zählerausgänge QA, QB, QC des *D 192* (D21) gelegt.
Den Zeichengenerator kann man auch in kombinierter RAM- bzw. PROM-Ausführung realisieren. Beispielsweise ließ sich der ASCII-Zeichensatz auf PROM abspeichern, und Graphikelemente in der oben geschilderten Art könnte man im RAM-Speicher programmieren.
Für den Taktsperreingang des *74165* gilt das in Abschnitt 5.3.1. Gesagte.

5.3.3. Vollgraphik

In diesem Fall kann jeder auf dem Bildschirm darstellbare Bildpunkt softwaremäßig angesprochen werden. Der Zeichengenera-

tor entfällt. Als Bildwiederholspeicher nutzt man einen RAM-Block mit entsprechend großer Speicherkapazität (z. B. 256×256 Bit). Hierzu wird entweder dem Fernsehinterface ein Bildwiederholspeicher zugeordnet oder ein Teil des Arbeitsspeichers des Computers genutzt. Im letzteren Fall liegen die Eingänge des Parallel-Serienwandlers dann direkt am Datenbus. Die Videoadressen (d. h. die Zählerausgänge zur zyklischen Adressierung) werden über Bustreiber an den Adreßbus gelegt. Die für den Speicherzugriff nötigen Steuersignale muß eine entsprechende Logik erzeugen. Der DMA-Vorgang, d. h. das Auslesen des Bildwiederholspeichers, liegt in der Vertikalhelltastphase. Das Vertikalaustastsignal ist gleichzeitig das $\overline{\text{BUSRQ}}$-Signal (in der Darstellung gemäß Bild 5.8 in invertierter Form). Die CPU kann damit nur in der Vertikalaustastphase arbeiten.
Mit dem Signal $\overline{\text{BUSAK}}$ = L werden die Adreßtreiber freigegeben und damit die aktuelle CRT-Adresse auf den Adreßbus gelegt. Verwendet man dynamische RAM, dann muß das Refresh aufrechterhalten werden.

5.4. Aufbau und Inbetriebnahme

Bei Inbetriebnahme des Bildschirminterface werden zunächst die RAM-Bestückung des Bildwiederholspeichers und die CPU-Baugruppe ausgeklammert.
Man beginnt mit dem Einstellen des Taktgenerators auf 9 MHz. Damit wäre bereits der einzig notwendige Abgleichvorgang an der Schaltung erledigt. Hierzu noch ein Hinweis: Der Bildpunkttakt wird mit dem Oszilloskop zweckmäßig durch exaktes Einstellen der Periode des Zeilensynchronimpulses mit T_H = 64 μs fein justiert.
Anschließend werden sämtliche Synchron- und Austastsignale und, wenn diese in Ordnung sind, das komplette BAS-Signal oszillographisch kontrolliert. Genügen diese Signale den angegebenen Bedingungen (vergleiche Bild 5.3, Bild 5.7, Bild 5.8), legt man die Multiplexerausgänge A0... A7 auf die entsprechenden Eingänge des Zeichenlatch. Auf dem Bildschirm muß jetzt der komplette Zeichensatz abgebildet sein (normal und invertiert).
Nach positivem Abschluß dieser Kontrollvorgänge werden die Verbindungen Multiplexer – Zeichenlatch beseitigt und der Bild-

wiederholspeicher mit den RAM-Schaltkreisen bestückt. Im Anschluß hieran verbindet man die Bildschirmplatine über den Systembus mit der CPU-Baugruppe und hat jetzt die Möglichkeit, ein geeignetes Testprogramm zu starten.
Beispielsweise kann der Zeichensatz auf den Bildschirm geschrieben werden. Das kleine Programm dazu ist im folgenden angegeben:

```
        LD HL, E800H
M1:     LD (HL), L
        LD L, (HL)
        INC HL
        LD A, H
        CMP F0H
        JRNZ M1
        HALT
```

Das Bildschirminterface funktioniert dann vorschriftsmäßig, wenn der vollständige Zeichensatz 6mal in steigender Reihenfolge der Zeichen gemäß ISO-7-Bit bzw. ASCII auf dem Bildschirm geschrieben wird.

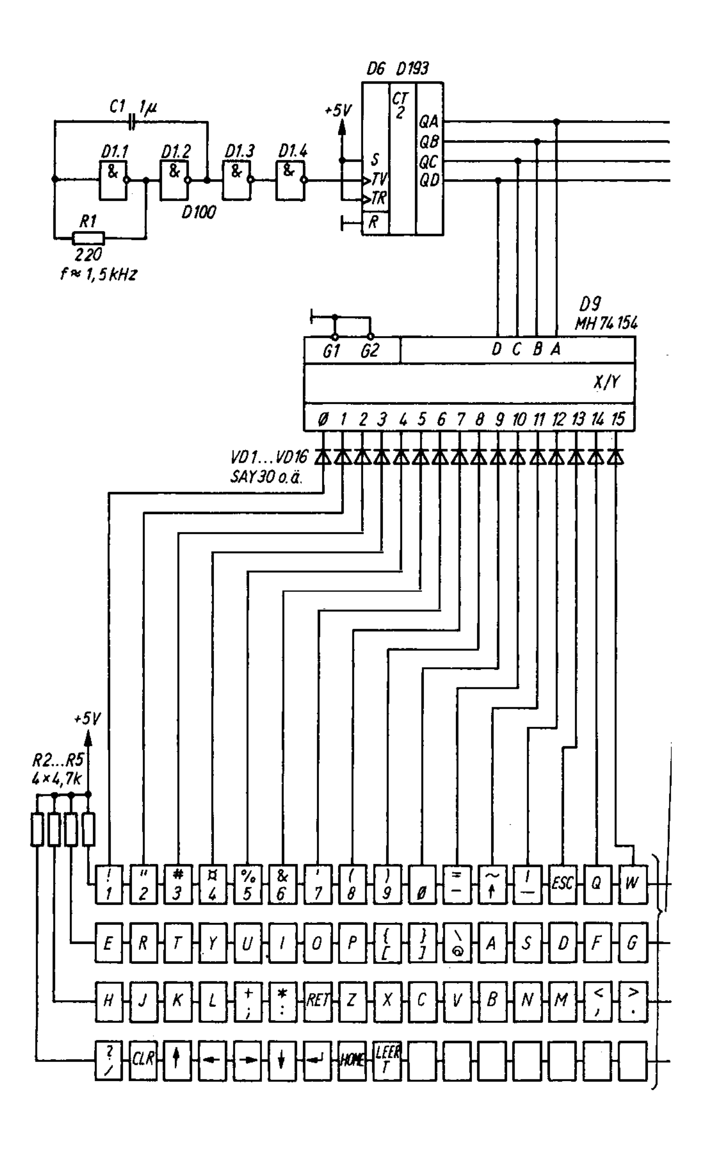
6. Alphanumerische Tastatur

Für den Dialog mit dem beschriebenen Mikrocomputer ist eine geeignete Tastatur die wichtigste Voraussetzung der Eingabe. Die im folgenden vorgestellte, einfach nachzubauende alphanumerische Tastatur läßt sich mit bis zu 64 Tasten auslegen. Im Gegensatz zu anderen denkbaren und auch üblichen Lösungen wird der jeweilige Tastaturcode hardwaremäßig erzeugt.

6.1. Schaltungsbeschreibung

Die Tastatur ist so konzipiert, daß beim Drücken einer Taste der entsprechende Tastaturcode auf dem Datenbus zur Verfügung gestellt wird. Dabei muß natürlich der Tastaturcode identisch mit dem auch im Bildschirminterface verwendeten Zeichencode gemäß ISO-7-Bit bzw. ASCII sein. Alle gewünschten Tastaturcodes werden auf einem PROM (*U 555*) gespeichert. Das Tastaturinterface sorgt nun dafür, daß beim Drücken einer Taste dem PROM gerade die Adresse bereitgestellt wird, auf der der dazugehörige Tastaturcode abgespeichert ist. Für die Meldung an das System über das Vorliegen eines gültigen Tastaturcodes steht ein Konsolstatussignal zur Verfügung.
Mit Hilfe eines Fertigmeldeimpulses kann man auch die softwaremäßige Übernahme des gültigen Tastaturcodes durch die CPU z. B. über nicht maskierbaren Interrupt ($\overline{NMI}$) einleiten.

Funktionsweise (Bild 6.1):
Ein Taktgenerator ($f \approx 1{,}5$ KHz) steuert den 4-Bit-Binärzähler *D 193* an, dessen Ausgänge dem 1. Schaltkreis *D 195* (D7) eines insgesamt 6-Bit-Adreßlatch zugeführt werden. Die Zählerzustände des *D 193* (D6) bilden damit nach Übernahme in den Latch die 4 niederwärtigen Adreßbit A0 . . . A3 für den Tastatur-PROM. Gleichzeitig wird aus ihnen mit Hilfe des 1-aus-16-Decoders *MH 74154* das Spaltenauswahlsignal für die Kontaktmatrix der Tastatur gewonnen. Die Decoderausgänge werden dabei durch die Dioden VD1 . . . VD16 geschützt.



C1 1µ
D1.1
D1.2
D1.3
D1.4
D100
R1
220
f ≈ 1,5 kHz
+5V
D6 D193
CT 2
S
TV
TR
R
QA
QB
QC
QD
D9
MH 74 154
G1 G2
D C B A
X/Y
Ø 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
VD1…VD16
SAY30 o.ä.
+5V
R2…R5
4×4,7k
ESC
CLR
RET
HOME
LEER

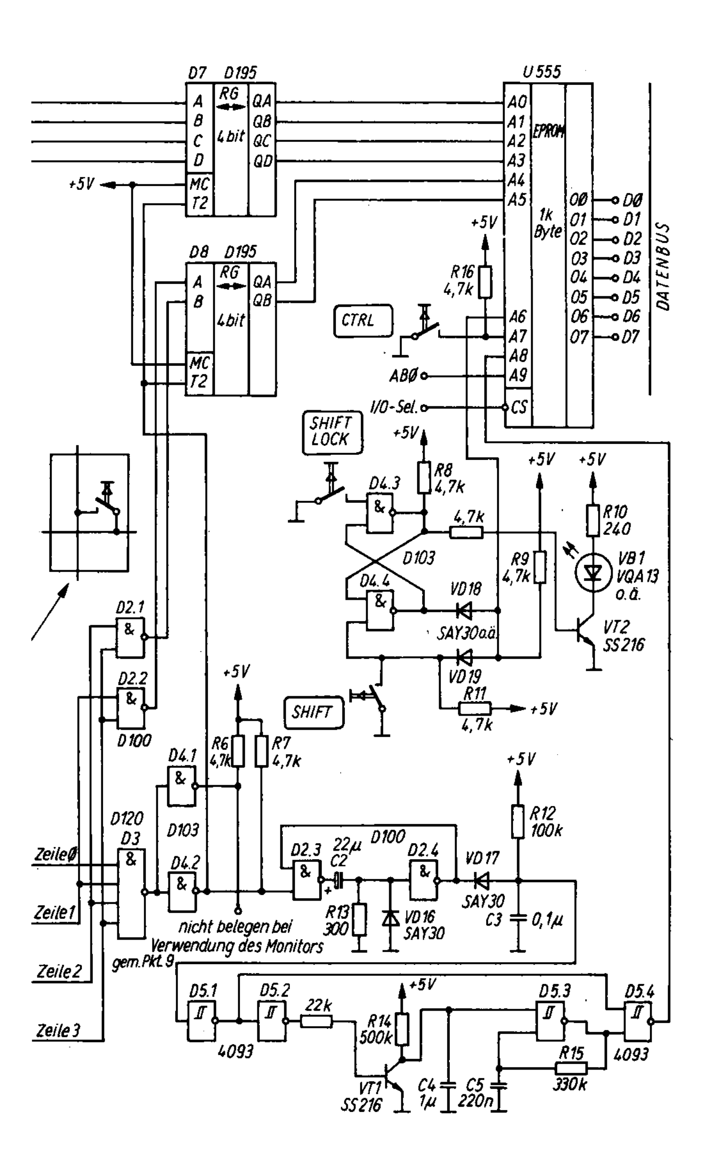
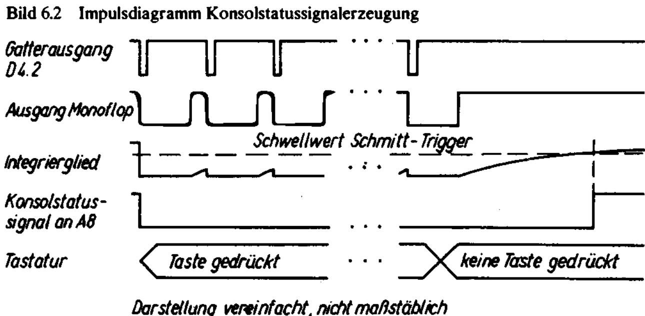


Bild 6.1 Alphanumerische Tastatur

Für den Fall, daß keine Taste gedrückt ist, liegen alle Zeilenpotentiale der Kontaktmatrix über den Widerständen R2 ... R5 auf H-Pegel. Beim Schließen eines Tastenkontaktes in der Zeile 0 wird über den jeweiligen Decoderausgang das Zeilenpotential kurzzeitig auf L gezogen. An den Gattern D3, D4.2 entsteht ein Übernahmeimpuls für die 2 Schaltkreise *D 195* des Adreßlatch (HL – Flanke an T2). Damit ist im 1. *D 195* (D7) der der gerade gedrückten Taste entsprechende Zählerzustand des *D 193* gespeichert und liegt als Adresse an den Eingängen A0 ... A3 des Tastatur-PROM vor. Die adressierte Speicherzelle stellt den gewünschten Tastaturcode am Datenbus zur Verfügung.
Der Latchimpuls kann am Ausgang von Gatter D4.1 als Fertigimpuls abgenommen werden. Gleichzeitig steuert er einen Monoflop (bestehend aus den Gattern D2.3, 2.4) an, der das Konsolstatussignal erzeugt. Dieses Signal wird nach Durchlaufen eines Integrierglieds über die Gatter D5.1, 5.4 des Schmitt-Trigger-Schaltkreises *4093* dem Adreßeingang A8 des PROM zugeführt. Dazu liegt im Normalfall der Ausgang von Gatter D5.3 auf H. Wenn eine Taste länger, als bei normaler Betätigung üblich, gedrückt wird ($t = 0{,}5$ s), erreicht die Spannung an $C4$ den Schwellwert des *Schmitt*-Triggers D5.3, der als Generator geschaltet ist. Der Generator schwingt mit etwa 15 Hz, und das Konsolstatussignal steht an A8 in ständiger Wiederholung bereit. Damit wiederholt sich auch die Übernahme des Tastaturcodes so lange, wie die Tastatur gedrückt ist.
Kontaktprellungen machen sich durch den Ausfall eines oder mehrerer Impulse am Gatter D4.2 bemerkbar. Wie aus dem Impulsver-

Bild 6.2 Impulsdiagramm Konsolstatussignalerzeugung



lauf in Bild 6.2 zu ersehen ist, wählt man die Zeitkonstante des Integriergliedes deshalb so, daß im genannten Fall der Schwellwert des nachfolgenden Schmitt-Triggers nicht erreicht wird. Damit ist eine ausreichende Prellunterdrückung gewährleistet.
Beim Betätigen einer Taste in den Zeilen 1 . . . 3 laufen prinzipiell die gleichen Vorgänge ab. Zusätzlich werden jedoch die der höheren Tastenwertigkeit entsprechenden Adreßbit A4 und A5 durch die Gatter D2.1, 2.2 erzeugt und im 2. *D 195* (D8) des Adreßlatch gespeichert.
Tabelle 6.1. enthält die sich ergebende Zuordnung der Tastenwertigkeiten, die der Bitbelegung A0 . . . A5 am PROM entspricht.
Das RS-Flip-Flop (D4.3, D4.4) ermöglicht die Umschaltung SHIFT bzw. SHIFTLOCK. Damit wird die bei Tastaturen übliche Zeitfunktion der Tasten (z. B. Groß- und Kleinbuchstaben) über den Adreßeingang A6 des Tastatur-PROM realisiert. Der Zustand SHIFTLOCK wird mit der LED VB1 angezeigt.
Mit der an den Adreßeingang A7 angeschlossenen CTRL-Taste lassen sich weitere Tastenfunktionen erzeugen, natürlich nur bei entsprechenden programmierten PROM.
Der Adreßeingang A9 wird mit der Adreßlinie AB0 des Systembusses verbunden und erfüllt eine Funktion im Zusammenhang mit der Konsolstatusabfrage.
Über den CS-Eingang wird die I/O-Portzuordnung der Tastatur realisiert. Dementsprechend verbindet man einen Ausgang des I/O-Portdecoders (in diesem Fall I/O-Port 08H) auf der CPU-Platine.
Prinzipiell lassen sich die Tasten in der Tastaturmatrix beliebig zuordnen. Es wird jedoch gefordert, daß auf der jeweils erzeugten PROM-Adresse der richtige Tastaturcode abgespeichert ist.

Tabelle 6.1. Zuordnung der Tastenwertigkeiten

Zeile der Kontaktmatrix	Tastenwertigkeit (A0 . . . A5)
0	0 . . . 15
1	16 . . . 31
2	32 . . . 47
3	48 . . . 63

Für die in Bild 6.1 gewählte Tastenzuordnung muß man den Tastatur-PROM *U 555* gemäß ISO-7-Bit-bzw. ASCII-Code entsprechend Tabelle 6.2. programmieren. In Tabelle 6.2. wurden neben den eigentlichen Tastaturcodes auch die zur Realisierung der Konsolstatusfrage notwendigen Randbedingungen berücksichtigt.

6.2. Aufbau und Inbetriebnahme

Der Aufbau einer alphanumerischen Tastatur erfordert eine sinnvolle, d. h. bedienungsgerechte, Tastenanordnung (ähnlich einer Schreibmaschinentastatur).
Bild 6.3 zeigt eine derartige Tastenanordnung, zugeschnitten auf das vorgestellte Mikrocomputersystem. Als Tasten eignen sich beispielsweise die im Amateurhandel erhältlichen Mikrotaster des VEB *Elektroschaltgerätewerk* Auerbach.
Die Tastatur wird mit einem Kabel an den Computer angeschlossen. Bringt man den Tastatur-PROM unmittelbar in der Tastatur unter, dann ergibt sich damit eine unerwünschte Verlängerung des Systembusses. Deshalb ist es sinnvoll, den Tastatur-PROM im Computereinschub, d. h. in Systembusnähe, zu plazieren.
Zur Inbetriebnahme – vorerst ohne PROM-Bestückung – wird zunächst die Taktfrequenz von etwa 1,5 kHz am Ausgang von Gatter D1.2 bzw. D1.4 mit dem Oszilloskop kontrolliert. Anschließend führt man eine oszillographische Signalkontrolle an folgenden Stellen der Schaltung durch:
- Zählerausgänge *D 193* (D6) einschließlich Eingänge *D 195* (D7), *MH 74154* (D9)
- Decoderausgänge *MH 74154*

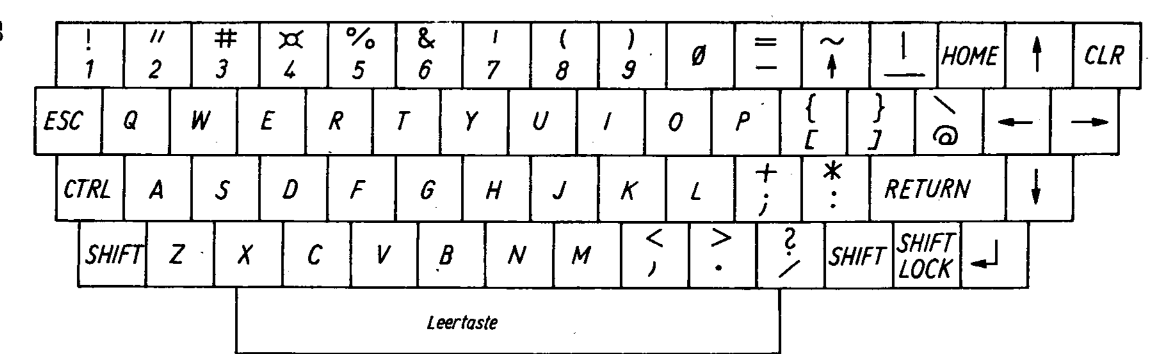
Das Decoderausgangssignal muß bei gedrückter Taste auch auf der jeweiligen Zeile erscheinen.
- Gatterausgang D3, D4.1, D4.2, Monoflop (D2.3, D2.4) Repeatschaltung (D5.2, D5.3), Adreßeingang A8 am Tastatur-PROM.

An diesen Punkten der Schaltung erscheint nur bei gedrückter Taste ein Signal. Damit schwingt also der Repeatgenerator, und am Gatterausgang D5.3 muß eine Rechteckspannung von etwa 15 Hz nachweisbar sein.
Sind alle Signale ordnungsgemäß vorhanden, steckt man den Tastatur-PROM auf und überprüft, ob nach dem Drücken einer Taste der richtige Tastaturcode an den Datenausgängen des PROM an-

Tabelle 6.2. HEXA-Tabelle Tastatur-PROM

Adresse HEXA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	01	02	03	04	05	06	07	08	09	00	1D	1E	1C	1B	11	17
010	05	12	14	19	15	09	0F	10	1B	1D	1C	01	13	04	06	07
020	08	0A	0B	0C	0B	0A	0D	10	18	03	16	02	0E	0D	1C	1E
030	1F	0C	0B	08	09	0A	1E	0E	20		– nicht belegt –					
040	01	02	03	04	05	06	07	08	09	00	1D	1E	1C	1B	11	17
050	05	12	14	19	15	09	0F	10	1B	1D	1C	01	13	04	06	07
060	08	0A	0B	0C	0B	0A	0D	1A	18	03	16	02	0E	0D	1C	1E
070	1F	0C	0B	0C	09	0A	1E	05	20		– nicht belegt –					
080	21	22	23	24	25	26	27	28	29	30	3D	7E	7C	1B	71	77
090	65	72	74	79	75	69	6F	70	7B	7D	5C	61	73	64	66	67
0A0	68	6A	6B	6C	2B	2A	0D	7A	78	63	76	62	6E	6D	3C	3E
0B0	3F	0C	0B	08	09	0A	1E	0E	20		– nicht belegt –					
0C0	31	32	33	34	35	36	37	38	39	30	2D	5E	5F	1B	51	57
0D0	45	52	54	59	55	49	4F	50	5B	5D	40	41	53	44	46	47
0E0	48	4A	4B	4C	3B	3A	0D	5A	58	43	56	42	4E	4D	2C	2E
0F0	2F	0C	0B	08	09	0A	1E	0E	20		– nicht belegt –					
100 . . . 1FF	alles 00/															
200 . . . 2FF	alles FF															
300 . . . 3FF	alles 00															

Anmerkung: Aus Gründen der Bedienungsfreundlichkeit des Monitorprogramms sind die Belegungen der Adreßbereiche 080H ... 0BFH und 0C0H ... 0FFH (= SHIFT/Normal) gegenüber ASCII bzw. ISO-7-Bit vertauscht.



Hinweis: Für die Tasten SHIFT, RETURN, Leertaste sollten möglichst 2 parallel geschaltete Mikrotaster verwendet werden.

Bild 6.3 Tastenanordnung für alphanumerische Tastatur

liegt. Gemäß ISO-7-Bit bzw. ASCII (siehe Anhang) muß sich z. B. für den Buchstaben A der Code 41 H ergeben, wenn nicht gleichzeitig die Taste SHIFT gedrückt ist (bzw. SHIFTLOCK nicht vorher gedrückt wurde). In dieser Art lassen sich alle Tastenfunktionen bzw. -codes kontrollieren.
Als praktisches Hilfsmittel für diese Kontrolle erweist sich die Verwendung von 8 LED mit entsprechenden Treibertransistoren. Notfalls erfüllt natürlich auch ein einfacher Vielfachmesser den gewünschten Zweck.
Das Zusammenspiel der Tastatur mit dem Computer kann mit einem kleinen Testprogramm bzw. dem in Abschnitt 9. behandelten Monitorprogramm überprüft werden.
Vor dem Start des im folgenden angegebenen Testprogramms sind der CS-Eingang des Tastatur-PROM auf das I/O-Port 08H und der Adreßeingang A9 auf die Adreßlinie AB0 des Systembusses zu legen. Der Bildschirm muß auf E800H selektiert sein.

```
M1:     LD HL, E7FFH
M2:     INC HL
        LD A, H
        CMP EEH
        JRZ M1
M3:     IN A, (09H)
        OR A
        JRNZ M3
M4:     IN A, (09H)
        OR A
        JRZ M4
        IN A, (08H)
        LD (HL), A
        JR M2
```

Bei richtiger Funktion der Tastatur werden die der jeweils gedrückten Taste entsprechenden Zeichen nacheinander, links oben beginnend, auf den Bildschirm geschrieben. Bei längerem Drükken einer Taste wirkt die Repeatfunktion. Solange die Taste gedrückt ist, wird ein Zeichen in ständiger Wiederholung ausgegeben.

7. Ein-/Ausgabebaugruppen

Der Computer kann bisher nur über die Tastatur und den Bildschirm Kontakt mit der Außenwelt aufnehmen. Mit geeigneten Ein-/Ausgabeschaltungen wird der Computer zu einem universell verwendbaren Gerät, mit dem sich auch die vielfältigsten Meß-, Steuer- und Regelaufgaben sowie eine Datenübertragung realisieren lassen. Einige Interfacetechniken sollen in diesem Abschnitt beschrieben werden.

7.1. Universelle I/O-Karte

Das Herz vieler Ein-/Ausgabeschaltungen ist eine universelle I/O-Leiterkarte. Die Leiterkarte enthält 2 PIO-Bausteine, 1 SIO-Baustein und 1 CTC-Baustein. Auf der Leiterkarte sind zusätzlich noch ein Kassetteninterface (siehe Abschnitt 8.), ein Port zum Abschalten des Computers und ein Tongenerator für Signalzwecke untergebracht.

Bild 7.1 und Bild 7.2 zeigen die Schaltung der Leiterkarte. Will man das Monitorprogramm aus Abschnitt 9. verwenden, dann müssen die IOSEL-Leitungen so mit dem zentralen I/O-Decoder verbunden werden, daß sich folgende Adressen ergeben (siehe auch Tabelle 3.2.):

ØØH Power-off-Port
Ø4H Tongenerator aus
Ø5H Tongenerator an

Ø8H Daten	Tastatur
Ø9H Status	Tastatur
ØCH Daten	PIO 1 Port A
ØDH Steuerwort	PIO 1 Port A
ØEH Daten	PIO 1 Port B
ØFH Steuerwort	PIO 1 Port B
10 H Daten	PIO 2 Port A
11H Steuerwort	PIO 2 Port A
12H Daten	PIO 2 Port B
13H Steuerwort	PIO 2 Port B

14H Kanal 0	CTC
15H Kanal 1	CTC
16H Kanal 2	CTC
17H Kanal 3	CTC
18H Daten	SIO Port A
19H Steuerwort	SIO Port A
20H Daten	SIO Port B
21H Steuerwort	SIO Port B

Die PIO 1 wird im System zum Anschluß eines Lochbandlesers genutzt, Port A überträgt die Daten, Port B die Steuersignale. Mit der PIO 2 stehen weitere 16 Ein-/Ausgabeleitungen zur Verfügung.
Zur seriellen Ein-/Ausgabe ist eine SIO vorgesehen. Die SIO wandelt 8-Bit-parallele in 1-Bit-serielle Informationen um. Somit sind nur sehr wenige Verbindungsleitungen zur Datenübertragung notwendig. Weiterhin arbeiten viele externe Geräte von Natur aus seriell, beispielsweise Fernschreiber, Floppys und Magnetbandspeicher. Ein Interface zur Datenaufzeichnung auf Magnetband wird in Abschnitt 8. beschrieben. Die Datenaufzeichnung auf Magnetband ermöglicht das SIO Port A.
Zum seriellen Informationsaustausch zwischen dem Computer und anderen Geräten ist eine V24-Schnittstelle vorgesehen (SIO Port B). Anpaßstufen sorgen für die Erzeugung der notwendigen Spannungspegel. Die V24-Schnittstelle arbeitet mit den Pegeln +12 V (0-Bit) und −12 V (1-Bit). Durch die Wahl dieser Pegel wird eine sichere Datenübertragung auch über größere Entfernungen garantiert. Außer den seriellen Sende- und Empfangsdaten (TxD und *RxD) werden noch folgende Steuersignale auf die Schnittstelle* geführt:

RTSB Sendeaufforderung	(Ausgang)
DTRB Datenstation bereit	(Ausgang)
DCDB Empfängerfreigabe	(Eingang)
CTSB Senderfreigabe	(Eingang)

Mit diesen Steuersignalen können verschieden schnelle Geräte miteinander synchronisiert werden. Beispielsweise läßt sich mit Hilfe des CTS-Eingangs das Senden von Daten verhindern, wenn das periphere Gerät die eingetroffenen Daten noch nicht verarbeitet hat. Umgekehrt kann das Signal RTS verhindern, daß das periphere Gerät weitere Daten sendet, solange der Computer die empfangenen Informationen noch nicht verarbeitet hat. Dazu muß

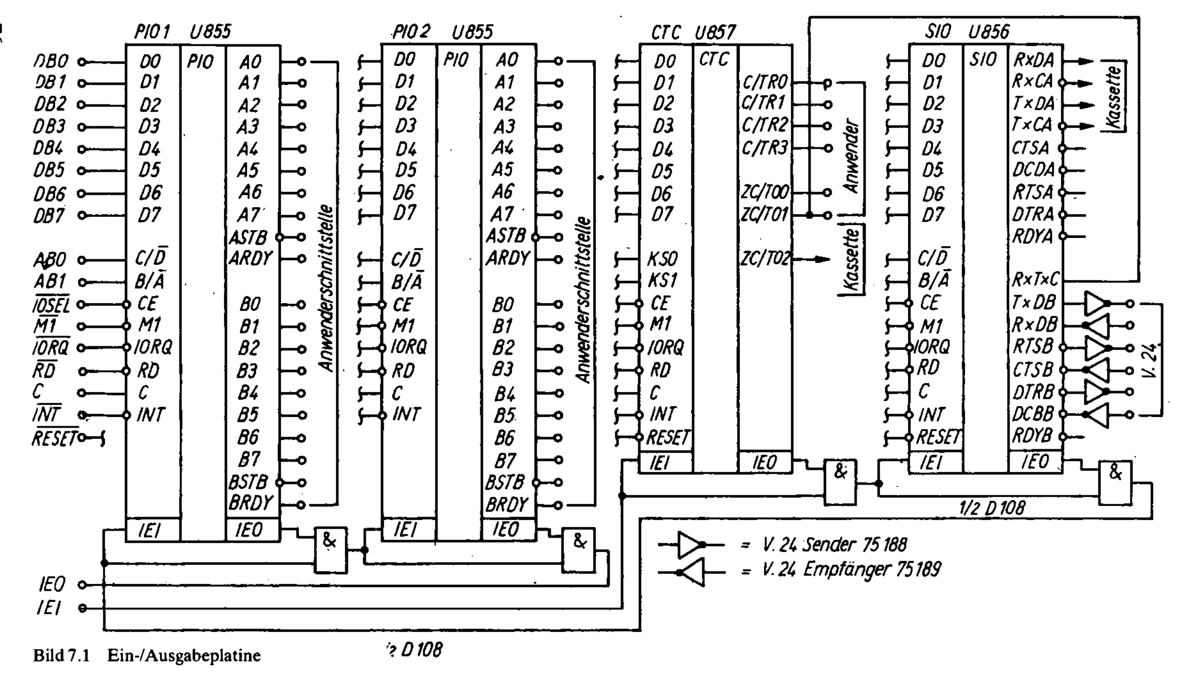


Bild 7.1 Ein-/Ausgabeplatine

man die Anschlußpunkte CTS und RTS des Computers und des peripheren Geräts kreuzweise miteinander verbinden. Die Daten werden mit genormten Geschwindigkeiten übertragen.
Üblich sind z. B. folgende Übertragungsraten (Baudraten, 1 baud = 1 Bit/s):
45.45, 50, 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200.
Die Sende- und Empfangstakte erzeugt ein CTC-Baustein. Durch entsprechende Programmierung des CTC können die verschiedenen Baudraten eingestellt werden (siehe auch Abschnitt 8.). Der CTC-Kanal 2 ist mit dem Kassetteninterface und der CTC-Kanal 1 mit dem Sende-/Empfangstakteingang $\overline{\text{RxTxCB}}$ des SIO-Schaltkreises verbunden. Die anderen beiden CTC-Kanäle stehen dem Anwender frei zur Verfügung.
Bei systemseitigem Ansteuern der I/O-Leiterkarte wurde auf den Einsatz von Treiberschaltkreisen verzichtet. Beim Aufbau eines sehr großen Systems wird jedoch der Einsatz von Treiberschaltkreisen (z. B. *8216*) empfohlen. Die AND-Gatter verringern die Einschwingzeiten der Interruptprioritätskette [2].
Auf der I/O-Karte fanden noch 2 kleine Schaltungen Platz, die in Bild 7.2 zu sehen sind. Die Schaltkreise D1, D2, D4.3 und N1 bilden einen Tongenerator, der mit dem Befehl OUT 5 angeschaltet und mit OUT 4 wieder abgeschaltet werden kann. Das JK-Flip-Flop D2 arbeitet als Frequenzteiler und als Tor. Das Rechtecksignal wird vom Transistor VT1 verstärkt und dem (hochohmigen) Lautsprecher zugeführt.
Wenn wie im Mustergerät die Netzzuschaltung mit einer Relaisselbsthalteschaltung erfolgt, so kann mit der 2. Schaltung der Computer softwaregesteuert abgeschaltet werden. Der Kontakt des Relais K unterbricht nach der Befehlssequenz OUT 0, OUT 0 und nach einer Zeit von etwa 18 s die Spannungszuführung des Netzrelais. Die Zeitverzögerung wird vom RC-Glied am Eingang des Triggerschaltkreises N2 bestimmt. Das Schieberegister D3.1 und D3.2 verzögern das Aufladen des Kondensators um einen OUT-Befehl. Mit IN0-Befehlen, $\overline{\text{RESET}}$ oder $\overline{\text{NMI}}$ kann die Abschaltlogik zurückgestellt werden. Der Transistor VT2 beschleunigt in diesen Fällen das Entladen des Kondensators.

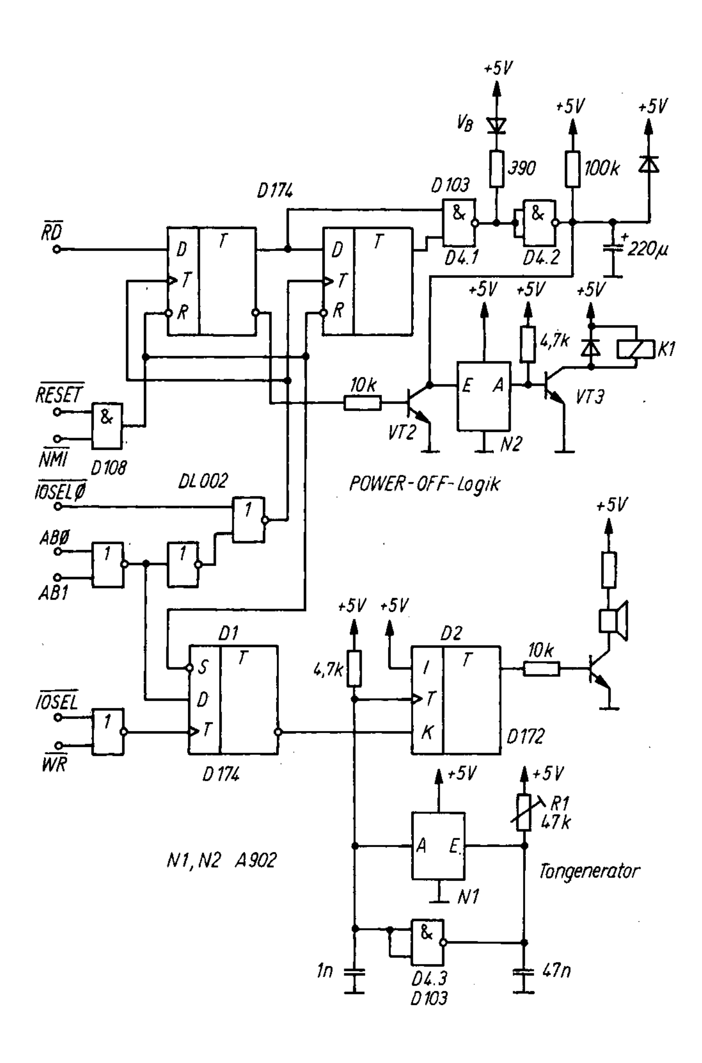


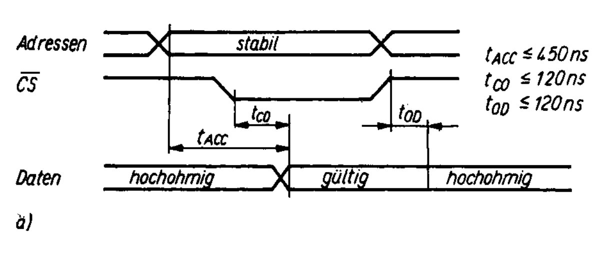
Bild 7.2 Power-off-Schaltung, Tongenerator

7.2. EPROM-Programmiergerät

Ein wichtiger Zusatz zum Computer ist das EPROM-Programmiergerät. Da man zukünftig auch höher integrierte EPROM-Schaltkreise erhalten wird, wurde es so entwickelt, daß sich nicht nur Speicherschaltkreise vom Typ *U555* bzw. *2708* programmieren lassen, sondern das gesamte Typenspektrum: vom „Kleinsten" *2704* bis zum *27256*. Die Anschlußbelegungen der verschiedenen EPROM-Typen sind im Anhang enthalten. Ein Vergleich der Pin-Belegungen zeigt, daß es nur wenige Unterschiede gibt. Bei EPROM-Schaltkreisen im 24poligen DIL-Gehäuse sind es die Pins 18 ... 21 und 24. Über diese Pins werden $\overline{CS}$, Programmierspannung und Programmierimpulse, Betriebsspannungen und/oder weitere Adreßbit zugeführt. EPROM-Schaltkreise ab *2764* befinden sich im 28poligen Gehäuse. Bei diesen Schaltkreisen kommen noch 4 Anschlüsse hinzu, für die gleiches gilt. Die genannten Signale werden über Codierstecker zugeführt. Mit Hilfe dieser Codierstecker wird die Hardware an den jeweiligen EPROM-Typ angepaßt.

7.2.1. Programmiervorschriften

Im unprogrammierten Zustand des EPROM-Schaltkreises sind alle Bit 1 (Ausgänge H-Pegel). Programmiert wird somit durch Einschreiben von 0 Bit in die adressierten Zellen. Eine programmierte 0 kann nicht mehr umprogrammiert werden.
Bild 7.3 verdeutlicht das Programmieren des *U555* an Hand des Impulsdiagramms. Den $\overline{CS}$/WE-Eingang legt man auf +12 V, anschließend die 1. Adresse und das zugehörige Datenwort (Ausgänge 0_0 ... 0_7) an den EPROM. Es folgt ein Programmierimpuls (0,1 ... 1 ms, 26 V). Danach wird die Adresse um 1 erhöht und der Vorgang wiederholt, bis alle Adressen abgearbeitet sind. Das muß N-mal für alle 1024 Adressen wiederholt werden. Die Anzahl der Programmierschleifen N hängt von der Breite des Programmierimpulses t_{pw} ab. Die integrale Programmierdauer je Speicherplatz beträgt 100 ms, so daß bei t_{wp} = 1 ms 100 Programmierschleifen notwendig sind (*Intel*-Programmiervorschrift).
Moderne EPROM lassen sich einfacher programmieren. Der Schaltkreis *2716* wird mit 50-ms-TTL-Impulsen programmiert. Er



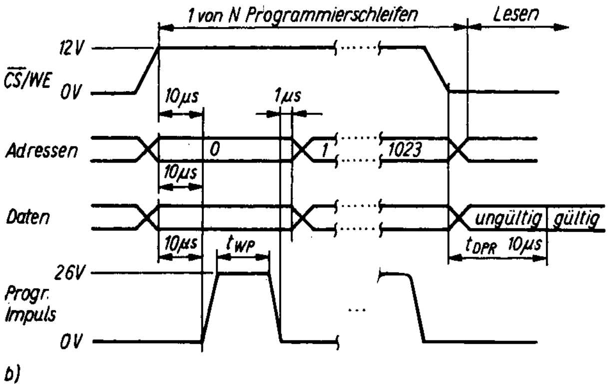


Bild 7.3 a – Impulsdiagramm *U555*, b – Impulsdiagramm *2708*

gestattet das Programmieren einzelner Speicherzellen und das Prüflesen während des Programmiervorgangs. Das Programmieren kann folgendermaßen ablaufen: Anlegen der Programmierspannung U_{pp} (25 V) an Pin 21, $\overline{OE}$ = H-Pegel, Anlegen Adresse und Datenbyte, Programmierimpuls (max. 55 ms) dem Eingang $\overline{CE}$ zuführen. Ein Prüflesen ist möglich, wenn der $\overline{OE}$-Eingang auf L gelegt wird, dabei muß U_{pp} nicht abgeschaltet werden. Bild 7.4 zeigt das Impulsdiagramm.
Der Schaltkreis *2732* wird ähnlich programmiert (Bild 7.5). Die Programmierspannung (25 V) wird dem Eingang $\overline{OE}/V_{pp}$ zugeführt, der Programmierimpuls (50 ms; L-Pegel) dem Eingang $\overline{CE}$.
Die genauen Programmiervorschriften sind unbedingt den Datenblättern des Herstellers zu entnehmen. Beispielsweise wird für den *U 555* eine integrale Programmierzeit von 50 ms je Speicherzelle

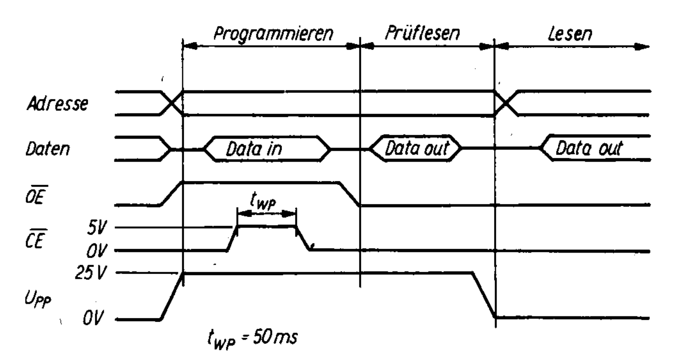


Bild 7.4 Impulsdiagramm 2716

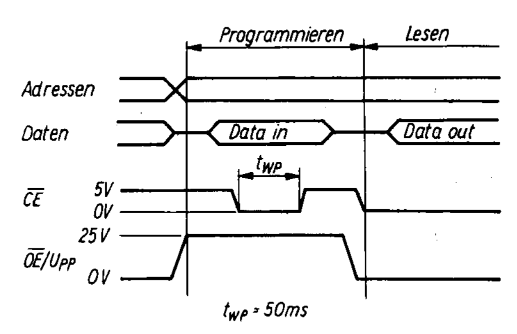


Bild 7.5 Impulsdiagramm 2732

[6], für den *Intel 2708* aber 100 ms [7] angegeben. Ein weiterer Hinweis in diesem Zusammenhang: Die EPROM-Schaltkreise *TMS 2716* und *i 2716* sind nicht kompatibel.

7.2.2. Die Schaltung

Bild 7.6 zeigt die Schaltung des Programmiergeräts. Ein PIO-Baustein überträgt die Daten und Steuersignale von und zum Programmiergerät. Die Daten werden über Port A übertragen. Um einen PIO-Baustein einzusparen, erzeugt man die Adressen hardwaremäßig. Dafür sind die Zähler D1 ... D4 vorgesehen. Der Zähler

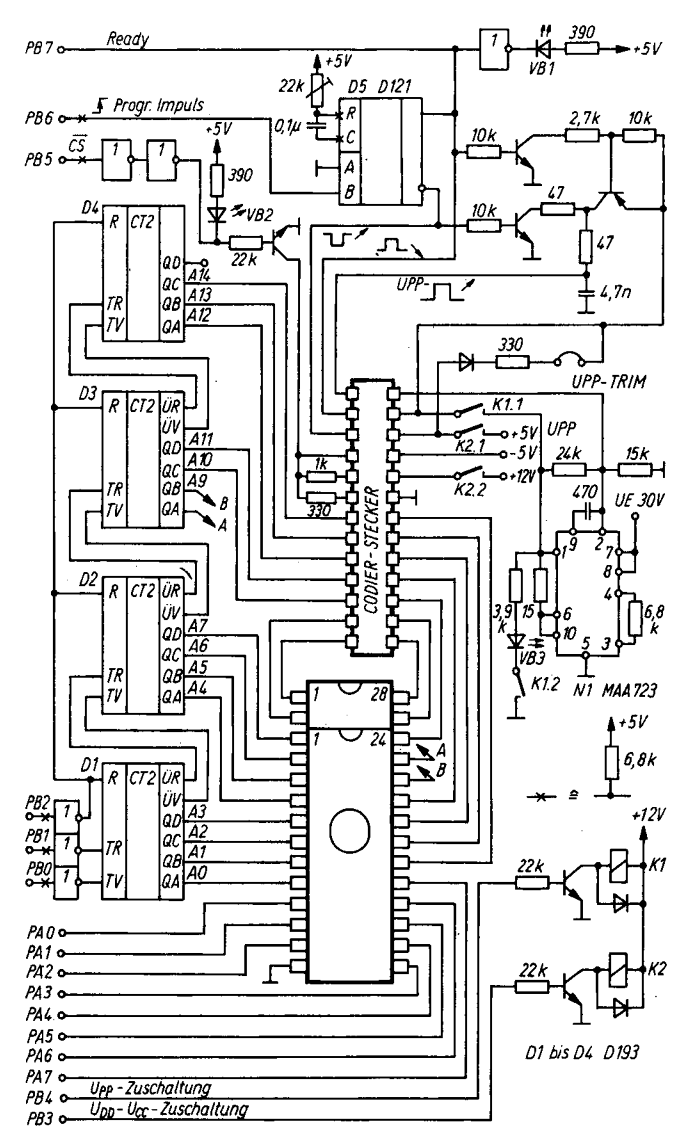


Bild 7.6 EPROM-Programmiergerät

D4 wird erst notwendig, wenn auch EPROM-Schaltkreise mit einer Kapazität $>$ 4 kByte programmiert werden sollen. Die Zählerausgänge sind mit dem EPROM-Sockel (A0 bis A9) und dem Codierstecker (A10 bis A14) verbunden.
Den Programmierimpuls von 1 ms Länge erzeugt der Monoflop D5. Diese Variante bietet gegenüber einer softwaremäßigen Lösung den Vorteil, daß auch im Havariefall die Impulsbreite 1 ms nicht überschritten und so ein Zerstören des EPROM verhindert wird. Um bei der Programmierung der Typen *2716, 2732, 2764* usw. die notwendige Programmierzeit von 50 ms je Speicherplatz zu erreichen, wird der monostabile Multivibrator 50mal angestoßen. Die Ausgänge Q und $\overline{\text{Q}}$ des Monoflop werden auf den Codierstecker geführt. Diese beiden Signale nutzt man bei den Schaltkreisen *2716, 2732* usw. als Programmierimpuls. Beim Programmieren der Typen *U 555, 2708, TMS 2716* muß der 1-ms-Impuls auf etwa 26 V verstärkt werden. Dafür ist der Verstärker mit den Transistoren VT1 . . . VT3 vorgesehen. Die 26-V-Impulse gelangen ebenfalls an den Codierstecker. Programmierimpulse zeigt die Leuchtdiode VB1 an.
Der Transistor VT4 arbeitet als Schaltstufe für das $\overline{\text{CS}}$/WE-Signal beim *U 555* bzw. *2708* oder für das $\overline{\text{OE}}$-Signal beim *2716* und *2732*. Den Zustand dieses Signals zeigt die Leuchtdiode VB2 an. Mit dem Relais K1 wird die Programmierspannung zugeschaltet. Die Leuchtdiode VB3 hat Kontrollfunktion. Das Programmiergerät enthält ein Netzteil für die Programmierspannung U_{pp}. Die Spannung stabilisiert man mit dem Schaltkreis N1 (*MAA 723*). Der Knotenpunkt der Widerstände *R1/R2* wird auf den Codierstecker geführt, um dort die genaue Programmierspannung mit einem Zusatzwiderstand festzulegen. Das Relais K2 schaltet die Betriebsspannung des EPROM-Schaltkreises softwaregesteuert zu.

7.2.3. Aufbau und Codierung

Die Programmiereinrichtung wurde als externes Gerät aufgebaut und läßt sich bei Bedarf über ein Kabel mit der PIO 2 verbinden. Das hat den Vorteil, daß die PIO 2 auch für andere Ein-/Ausgabeaufgaben genutzt werden kann.
Als EPROM-Fassung sollte ein 28poliger Nullkraftsockel eingesetzt werden. Ein 26poliger direkter Steckverbinder nimmt die Co-

dierstecker auf. Sie bestehen aus 30 mm × 35 mm großen doppelt kaschierten Leiterplatten. Bild 7.7 zeigt die Anschlußbelegung der Codiereinrichtung. Bild 7.8 bis Bild 7.10 zeigen die Beschaltung der Pins 18 . . . 21 sowie die Verdrahtung der Codierstecker für die EPROM-Typen *U 555 (2708)*, *2716* und *2732* (vergleiche auch Bild 7.3 bis Bild 7.5).

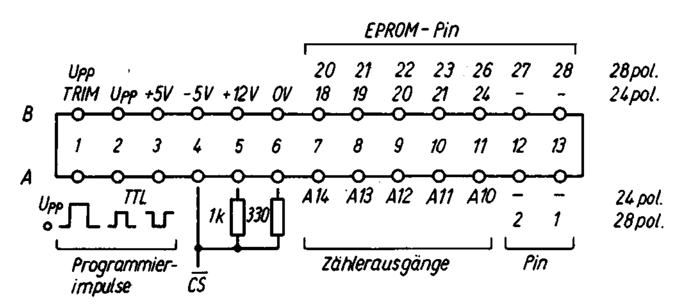


Bild 7.7 Anschlußbelegung der Codiervorrichtung

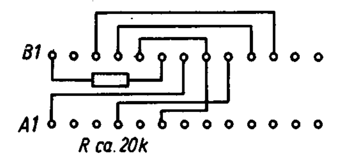


Bild 7.8 Codierstecker für *U555 (2708)*

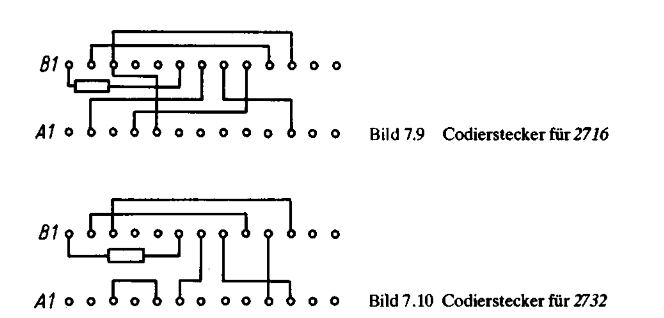


Bild 7.9 Codierstecker für *2716*

Bild 7.10 Codierstecker für *2732*

Die Verbindung des Programmiergeräts mit dem Computer übernimmt die PIO 2. Dabei sind die beiden Ports wie folgt belegt:

Port A Daten
Port B 0 Vorwärtszählen (positiver Impuls)
        1 Rückwärtszählen (positiver Impuls)
        2 Rücksetzen des Zählers (negativer Impuls)
        3 Zuschalten U_{cc} und U_{DD} (H)
        4 Zuschalten U_{pp} (H)
        5 $\overline{CS}$/WE bzw. $\overline{OE}$ (H = Lesen)
        6 Programmierimpuls (L/H-Flanke)
        7 Fertigsignal (H für die Dauer des Impulses)

7.2.4. Ein Beispielprogramm

An einem BASIC-geschriebenen Programm soll das Programmieren eines *U 555* demonstriert werden. Dabei wird abweichend von der *Intel*-Vorschrift programmiert [8], [9].

```
 10   PRINT
 20   PRINT „U 555/2708 PROGRAMMER“
 30   PRINT
 60   DA = 16 : CA = 17 : DB = 18 : CB = 19 : REM
      Ports festlegen
 70   OUT CB, & CF : OUT CB, & 80 : REM Bitmode, Bit
      7 Eingabe
 80   OUT CA, & 7F : REM Byteeingabe
 90   OUT DB, & 21 : REM Spannungen aus, Zählerreset.
100   INPUT „LESEN (1) PROG (2) VERGLEICHEN (3)
      LÖSCHTEST (4)“; M
110   IF M > 4 THEN END
120   ON M GOSUB 250, 320, 550, 140
130   GOTO 90
140   REM Löschkontrolle
150   OUT CA, & 7F : REM Byteeingabe
160   OUT DB, & 29 : OUT DB, & 2D : R
170   F = 0 : REM Flag
180   FOR I = 1 TO 1024 : IF INP (DA) <> & FF THEN F =
      1 : I = 1024
190   OUT DB, & 2c : OUT DB, & 2D : REM weiterzählen
200   NEXT I
```

```
210   PRINT „EPROM“; : IF F = 1 THEN PRINT „NICHT“;
220   PRINT „GELÖSCHT“ : RETURN
250   REM Dublizieren des EPROM-Inhalts ins RAM
260   INPUT „ADRESSE“; A
270   OUT CA, & 7F : REM Byteeingabe
280   OUT DB, & 29 OUT DB, & 2D : REM Reset
290   FOR I = A TO A + 1023 : POKE I, INP (DA)
300   OUT DB, & 2c : OUT DB, & 2D : REM weiterzählen
310   NEXT I : RETURN
320   REM programmieren
330   INPUT „ADRESSE“; A : GOSUB 150 : IF F = 1 THEN
      INPUT „WEITER (J/N)“; W$ I : F W$ <> „J“ THEN
      RETURN
340   PRINT : Z = Ø : REM Zykluszähler = 0
350   N = 2 : GOSUB 430 : REM 2 Programmierversuche
360   REM Prüflesen
370   OUT CA & 7F : OUT DB, & 2D : OUT DB, & 29 : OUT
       DB, & 2D
380   F = Ø : FOR I = A TO A + 1023 : IF INP (DA) <>PEEK
      (I) THEN F = 1 : I = A + 1024
390   OUT DB, & 2C : OUT DB, & 2D : NEXT I
400   IF F = 1 AND Z < 32 THEN 350
410   IF F = 1 THEN PRINT „NICHT PROGRAMMIERT“;:
      GOTO 520
420   PRINT : N = Z/2 : GOSUB 430 : GOTO 520 : REM
      Sicherheitszyklen
430   OUT DB, & 29 : OUT DB, & 19 : OUT DB, & 1D : REM
       Zählerreset und Spannungen anlegen
440   OUT CA, & F : REM Byteausgabe
450   FOR I = 1 TO N : Z = Z + 1 : PRINTCHRs (&B);
      USING „PROG ZYKLUS # # #“; Z
460   OUT DB, & 19 : OUT DB, & 1D : REM Zähler
      rücksetzen
470   FOR I = A TO A + 1023 : OUT DA, PEEK (I)
475   REM Programmierimpuls erzeugen und weiterzählen
480   OUT DB, & 5D : OUT DB, & 1D : OUT DB, & 1C :
      OUT DB, & 1D
490   NEXT I, J : RETURN
500   REM Auf Fertigsignal (Bit 7) wird nicht gewartet,
      weil Schleifenlaufzeit > 1 ms
```

```
510   OUTCA&7F:REMByteeingabe
520   OUTDB,&2D:REMAufLesenschalten
525   REMPrüflesen,anschließendSignalton
530   GOSUB560:OUT5,Ø:FORI=1TO500:NEXT
      I:OUT4,Ø
540   RETURN
550   INPUT„ADRESSE“;A
560   OUTCA,&7F:REMByteeingabe
570   OUTDB,&29:OUTDB,&2D:REMZähler
      rücksetzen
580   E=Ø:FORI=ATOA+1023
590   IFINP(DA)<>PEEK(I)THENE=E+1
600   OUTDB,&2C:OUTDB,&2D:REMweiterzählen
610   NEXTI:PRINTE;„FEHLER“:RETURN
```

7.3. Analogschnittstellen

Will man mit dem Computer auch Analogsignale oder Meßwerte erfassen und verarbeiten, so sind geeignete Analog/Digital-Wandler und Digital/Analog-Wandler erforderlich. In diesem Abschnitt werden einige Analogschnittstellen beschrieben.

7.3.1. Digital/Analog-Wandler

International gibt es eine große Anzahl hybrider und monolithischer D/A-Wandlerbausteine. Ein besonders günstiger Baustein ist der sowjetische CMOS-Wandler *K 572 П*A1. Es handelt sich bei diesem Schaltkreis um einen TTL-kompatiblen multiplizierenden 10-Bit-Wandler mit einer Einschwingzeit von 500 ns. Die Betriebsspannung kann 5 . . . 15 V (P_{vmax} = 20 mW), die Referenzspannung maximal ± 10 V betragen. Bild 7.11 zeigt das Funktionsprinzip des *K 572 П*A1 und Bild 7.12 die Standardbeschaltung. Die Ausgangsspannung beträgt

$$U_A = U_{ref} \sum_{i=1}^{n} \frac{Si}{2n - i + 1.}$$

Dabei gilt: Si = 1 bei H und Si = Ø bei L am Digitaleingang Ei und n = 10.
Bild 7.13 zeigt die Beschaltung des D/A-Wandlers für bipolare

Bild 7.11 D/A-Wandler *K 572 П A1*

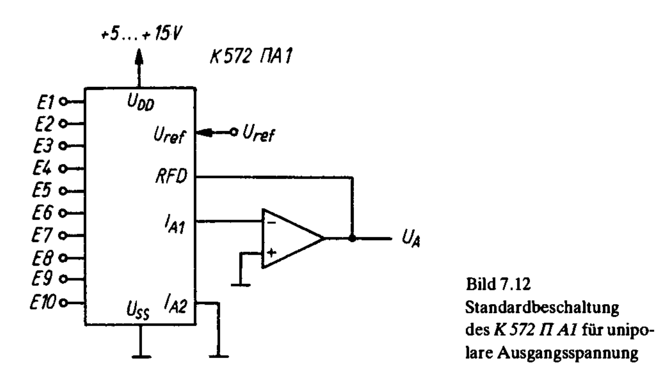


Bild 7.12 Standardbeschaltung des *K 572 П A1* für unipolare Ausgangsspannung

Ausgangsspannung. Die erreichbare Umsetzgeschwindigkeit ist von der Slew-Rate der verwendeten Operationsverstärker abhängig. Der A/D-Wandler *K 572 П* A1 gestattet einfache und elegante Problemlösungen. Er läßt sich beispielsweise in den Gegenkopplungszweig eines Verstärkers schalten, um eine digitale Verstär-

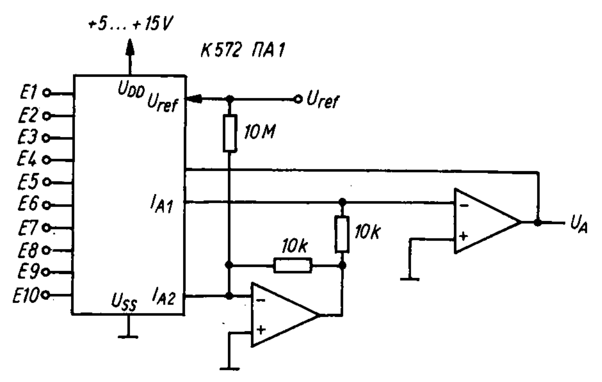


Bild 7.13 Standardbeschaltung des *K 572 П A1* für bipolare Ausgangsspannung

kereinstellung zu ermöglichen, oder man verwendet ihn als digital steuerbaren Spannungsteiler. Dabei können von dem D/A-Wandler auch Wechselspannungen bis zu $U_{ss} = 20$ V verarbeitet werden. Der Wandler läßt sich über einen PIO-Baustein an den Computer ankoppeln. Sind keine speziellen D/A-Wandlerschaltkreise verfügbar, so bieten sich auf der Grundlage von R-2R-Netzwerken verschiedene Lösungsvarianten an. Bild 7.14 und Bild 7.15. zeigen

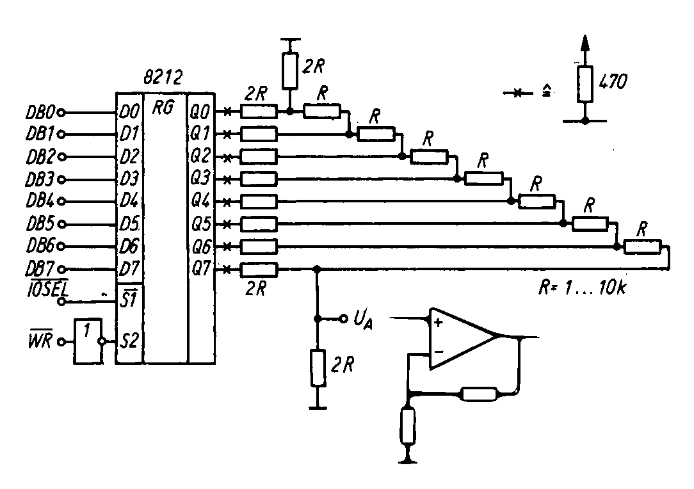


Bild 7.14 D/A-Wandler mit *8212*

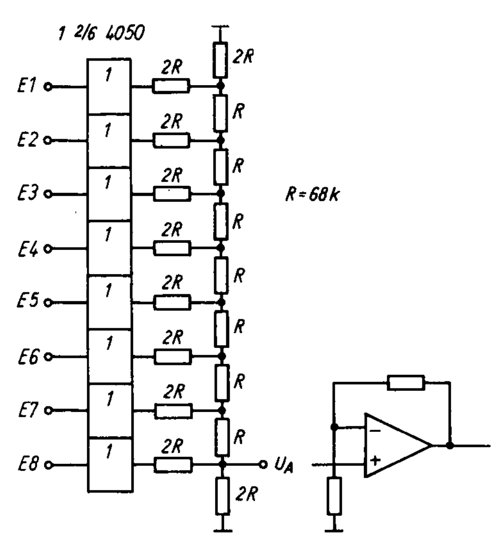


Bild 7.15 D/A-Wandler mit CMOS-Schaltkreisen *4050*

2 einfache Schaltungen. Die Ausgangsstufen der Schaltkreise arbeiten als Spannungsschalter. Die Betriebsspannung der Schaltkreise ist gleichzeitig die Referenzspannung. Die Genauigkeit dieser D/A-Wandler hängt von der Toleranz der Widerstände und von der Charakteristik der Spannungsschalter ab. Die Durchlaßwiderstände der Schalter müssen klein gegenüber R, die Sperrwiderstände groß gegenüber R sein, damit die Fehlerströme im Netzwerk klein genug bleiben. Wenn keine hohen Anforderungen an die Genauigkeit gestellt werden, ist die Schaltung in Bild 7.14 für viele Anwendungen geeignet. Der Schaltkreis *8212* arbeitet als Port und als Spannungsschalter. Günstigere Eigenschaften hinsichtlich der Genauigkeit hat die Schaltung in Bild 7.15. Hier werden die Spannungsschalter durch CMOS-Leistungstreiber realisiert. Die Widerstände sollten Metallschichttypen mit einer Toleranz 1 % aus einer Fertigungscharge sein. Die Ausgangsspannung kann mit der bereits angegebenen Gleichung berechnet werden ($U_{Ref} = U_{cc}$, n ≙ Schaltung).
In Anwendungsfällen, bei denen es nicht so sehr auf hohe Umsetzgeschwindigkeiten ankommt, bieten sich D/A-Wandler an, die

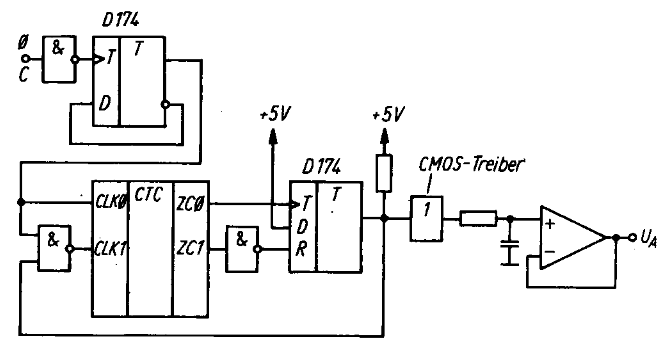


Bild 7.16 D/A-Wandler mit CTC-Baustein

durch die Pulsbreitenmodulation und anschließende Integration sehr genau umsetzen. Diese D/A-Wandler haben den Vorteil, daß sie sich ohne Präzisionsbauelemente aufbauen lassen [10]. Bild 7.16 zeigt einen 8-Bit-Digital/Analog-Umsetzer mit einem CTC-Baustein (siehe auch [2]). Beide CTC-Kanäle arbeiten im Zählermode. Der CTC-Kanal 0 erzeugt den Referenztakt (TC = 256). Der Kanal 1 wird mit dem zu wandelnden Wert geladen.
Mit 2 Digital/Analog-Wandlern (z. B. Bild 7.14) ist der Aufbau eines einfachen Oszillographen-Interface möglich. Bei dieser Anwendung wird der eine DAC-Ausgang mit dem X-Eingang, der andere DAC-Ausgang mit dem Y-Eingang des Oszillographen verbunden. Hat der Oszillograph einen Z-Eingang, so kann mit einem weiteren D/A-Wandler (geringerer Auflösung) auch eine Helligkeitsmodulation durchgeführt werden. Mit dieser Konfiguration ist eine einfache graphische Ausgabe möglich. Mit einem geeigneten Analog/Digital-Wandler läßt sich der Computer dann als NF-Speicheroszilloskop betreiben (bis einige hundert Hertz).

7.3.2. Analog/Digital-Wandler

Mit Hilfe der beschriebenen Wandler ist auf einfache Weise eine Analog/Digital-Wandlung möglich. Bild 7.17 und Bild 7.18 zeigen 2 mögliche Schaltungsvarianten, wobei die Schaltung in Bild 7.18 gleichzeitig 8 Analogkanäle erfassen kann. Das Signal wird soft-

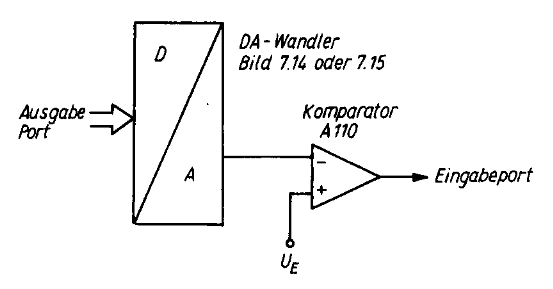


Bild 7.17 1-Kanal-A/D-Wandler

Bild 7.18 8-Kanal-A/D-Wandler

waregesteuert nach dem Verfahren der sukzessiven Approximation oder durch softwaremäßiges Nachbilden eines Vor-/Rückwärtszählers umgesetzt. Ein Komparator vergleicht die Ausgangsspannung des D/A-Wandlers mit der Eingangsspannung. Der Computer überwacht den Komparatorausgang. Je nach Pegelzustand des Komparators verändert die A/D-Software das Digitalwort, das anschließend an den D/A-Wandler ausgegeben wird. Ein Schaltungsbeispiel mit Vor- und Rückwärtszählern (Bild 7.19) soll verdeutlichen, wie man vorgeht. Die Ausgänge der Zähler D1 und D2 sind mit einem D/A-Wandler verbunden. Die Ausgangsspannung des D/A-Wandlers wird vom Komparator N1 mit der umzusetzenden Eingangsspannung verglichen. Das Komparatorausgangssignal, auf ein D-Flip-Flop geführt, bestimmt die Zählrichtung. Bei veränderter Eingangsspannung wird, je nach Vorzeichen der Änderung, vor- oder rückwärts gezählt, bis die Eingangsspannung und DAC-Ausgangsspannung gleich sind. Die Umsetzzeit ist

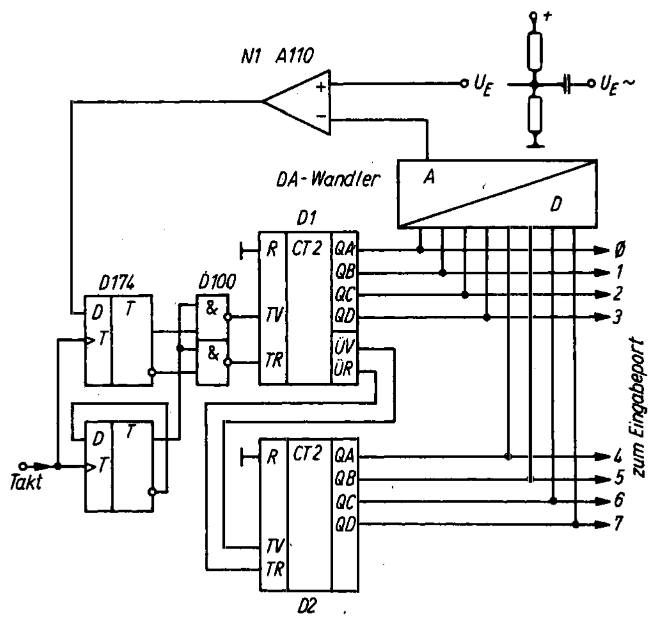


Bild 7.19 A/D-Wandler mit Vor-/Rückwärtszähler

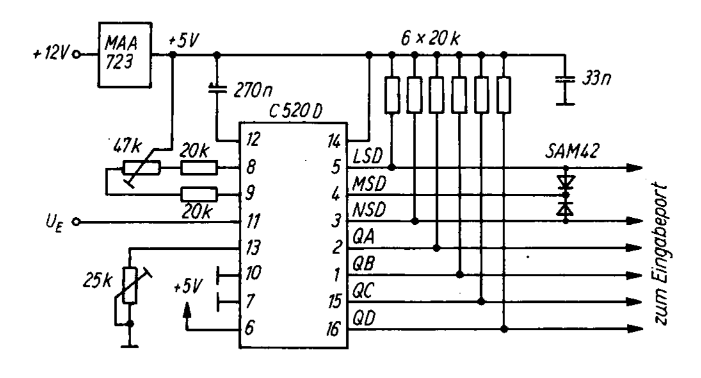


Bild 7.20 A/D-Wandler mit *C520*

von der Größe der Änderung und der Taktfrequenz abhängig. Bei der Wahl der Taktfrequenz muß man die Einschwingzeiten des D/A-Wandlers und des Komparators berücksichtigen. An den Zählerausgängen kann das Digitalwort abgenommen und einem Eingabeport zugeführt werden. In vielen Fällen ist es zur Verringerung von Umsetzfehlern notwendig, dem Komparator eine Sample & Hold-Baugruppe vorzuschalten, die die Eingangsspannung während der Umsetzperiode konstant hält. Mit derartigen A/D-Wandlern lassen sich, vorausgesetzt, die Taktfrequenz kann genügend hoch gewählt werden, auch NF-Signale digitalisieren. Dann sind z. B. Anwendungen wie das oben erwähnte Speicheroszilloskop oder eine digitale Tonverarbeitung in der Musikelektronik möglich.

Für die Anwendungen, bei denen es nicht auf eine hohe Umsetzrate ankommt, bieten sich integrierte A/D-Wandler an. Bild 7.20 zeigt eine aufwandarme Schaltungsvariante. Die A/D-Umsetzung übernimmt der für Schalttafelinstrumente vorgesehene Schaltkreis *C520*. In [11] wurde auch die notwendige Software veröffentlicht, so daß hier nicht auf weitere Einzelheiten eingegangen werden soll.

8. Kassetteninterface

Als externe Daten- und Programmspeicher setzt der Amateur vorwiegend Lochbänder und Magnetbänder ein. Das Magnetband ist ein preiswerter Datenspeicher hoher Kapazität. Im folgenden wird ein Kassetteninterface beschrieben, das die Aufzeichnung und die Wiedergabe von Daten und Programmen mit einem handelsüblichen Kassettenrecorder ermöglicht.

8.1. Aufzeichnungsverfahren

Ein Magnetbandgerät hat näherungsweise eine Bandpaßcharakteristik. Gleichspannungen, die bei einer langen Folge von L-Bit oder H-Bit auftreten, lassen sich nicht aufzeichnen. Digitale Daten kann man mit einem Audiokassettengerät nur aufzeichnen, wenn sie codiert sind. Das Codierungsverfahren sollte dabei den Übertragungseigenschaften des Speichermediums gut angepaßt sein. Einige Möglichkeiten der Codierung werden nachfolgend kurz aufgezeigt.
Ein sehr einfaches Verfahren ist die Amplitudenmodulation. Die Datenbit tasten einen NF-Träger von etwa 2 kHz. Bild 8.1 zeigt das modulierte Signal. Die Information gewinnt man durch Gleichrichten und Sieben des wiedergegebenen Signals zurück. Eine praktische Realisierung wurde in [2] veröffentlicht. Nachteil dieses Verfahrens ist die nur geringe Datenübertragungsrate von (meist) nur 110 baud (110 Bit/s). Die Aufzeichnung eines Blocks von 1 kByte Länge dauert etwa 102 s.
Auch die Frequenzmodulation ist zum Codieren geeignet. Beispiele dafür sind das KIM-Interface und der Kansas-City-Standard. Bild 8.2 zeigt den Kansas-City-Standard. Ein L-Bit besteht

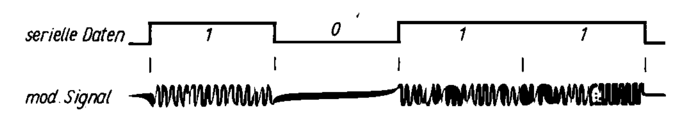


Bild 8.1 Aufzeichnung mittels Amplitudenmodulation

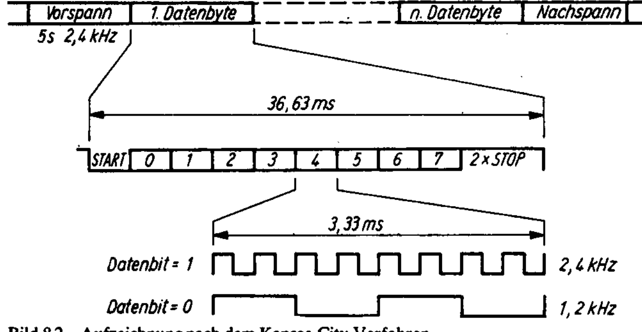


Bild 8.2 Aufzeichnung nach dem Kansas-City-Verfahren

aus 4 Schwingungen (1,2 kHz), ein H-Bit aus 8 Schwingungen (2,4 kHz). Moduliert wird z. B. mit einem FSK-Generator. Die Demodulation kann mit einem Impulsbreitendiskriminator durchgeführt werden. Ein Demodulator wurde in [5] beschrieben. Der Kansas-City-Standard ermöglicht eine recht sichere Datenaufzeichnung. Allerdings beträgt die Übertragungsgeschwindigkeit nur 300 baud, so daß ein Kansas-City-Interface für die Aufzeichnung großer Datenmengen ebenfalls nicht geeignet ist. Das Aufzeichnen eines 1-kByte-Blocks dauert etwa 38 s.
Weitaus höhere Datendichten erreicht man mit den verschiedenen PCM-Verfahren. Im Beispiel wurde Phase-Encoding (Richtungstaktschrift) verwendet. Bild 8.3 zeigt den Code sowie die Modula-

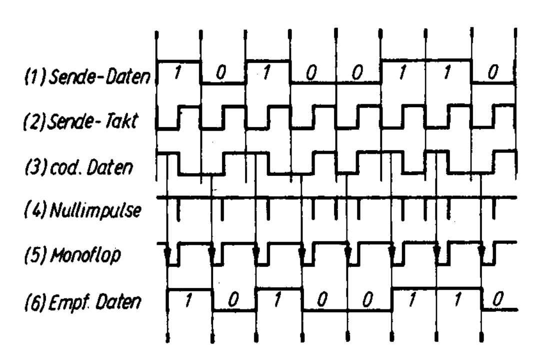


Bild 8.3 Aufzeichnung nach dem Phase-Encoding-Verfahren

tion und Demodulation. Eine Exklusiv-Oder-Verknüpfung von Takt (1) und Daten (2) bewirkt die Modulation. Die Impulsfolge (3) zeigt die codierten Daten: Ein H-Bit wird durch einen H/L-Sprung und ein L-Bit durch einen L/H-Sprung in der Mitte des Bit gekennzeichnet. Bei aufeinanderfolgenden gleichen Bit entstehen redundante Flußwechsel, die bei der Rückgewinnung der Information von den eigentlichen Flußwechseln unterschieden werden müssen. Das wird durch die Aufzeichnung von Synchronzeichen am Beginn eines Datenblocks erreicht. Bei der Wiedergabe gelangt das vom Kassettenrecorder kommende Signal an einen Nulldurchgangsdetektor. Die Nulldurchgänge (4) triggern einen monostabilen Multivibrator, der nach 3/4 der Bitlänge wieder zurückkippt (5). Mit der entstehenden Flanke wird die Information in das Empfangsschieberregister (6) geschoben.
Das Kassetteninterface arbeitet mit einer Bitrate von 4800 baud. 1 kByte läßt sich in nur 2,25 s aufzeichnen. Das von den Autoren realisierte Interface wird mit dem Kassettenrecorder *Geracord GC-6020* portable betrieben. Dieses Gerät verfügt über eine genau arbeitende Bandzähluhr, die das schnelle Auffinden von Datenaufzeichnungen ermöglicht. Auch andere Geräte wie *MIRA, MK 21* und *ZK 246* wurden an das Interface angeschlossen. Die Fehlerrate war bei Einsatz von hochwertigem Bandmaterial sehr gering. Treten bei der Wiedergabe dennoch Lesefehler auf, so sorgt die in Abschnitt 9. beschriebene Software für eine selbständige Fehlerkorrektur. Somit steht ein zuverlässiger, schneller und billiger Massenspeicher zur Verfügung.

8.2. Realisierung

Mit dem Interface lassen sich unterschiedliche Baudraten realisieren. Der etwas höhere Bauelementeaufwand bringt den zusätzlichen Vorteil, daß das Interface ohne komplizierten Abgleich in Betrieb genommen werden kann. Bild 8.4 zeigt die Schaltung des Kassetteninterface. Zentrale Bausteine sind die SIO *U 856* und der CTC *U 857* (siehe Abschnitt 7.1.). Das Kassetteninterface benutzt das Port A der SIO. Port A hat getrennte Eingänge für den Sendetakt (TxC) und den Empfangstakt (RxC), was für die Funktion der Schaltung nötig ist. Kanal 2 der CTC erzeugt den Takt. Gatter D1.3 arbeitet als Takttreiber für den Takt T.

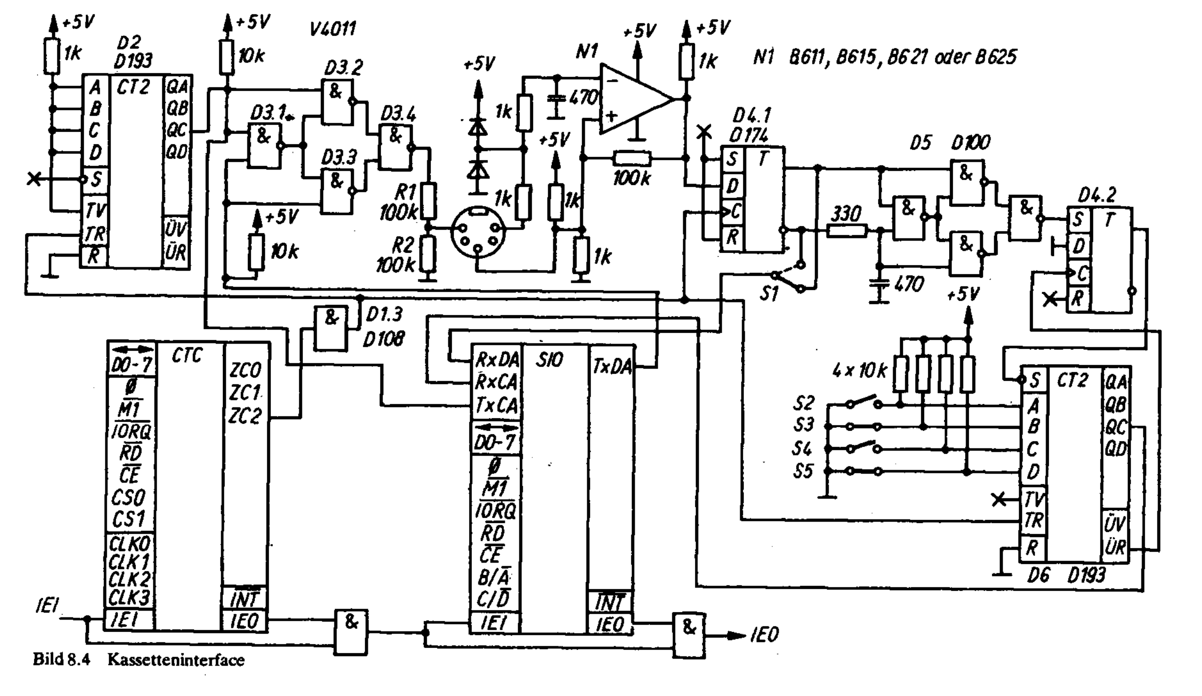


Bild 8.4 Kassetteninterface

Der Modulator besteht aus dem Zähler D2 (*D 193*) und einem Exklusiv-Oder, das aus den Gattern D3.1 ... D3.4 gebildet wird. Der Zähler D2 teilt den Takt T durch 8. An QC des Zählers wird der Sendetakt TxCA abgenommen und der SIO zugeführt. Die Baudrate entspricht der Frequenz des Sendetaktes TxCA. Jede fallende Flanke des Sendetaktes schiebt 1 Bit aus dem Senderegister der SIO. Dieses Bit wird mit dem Takt TxCA exklusiv-oder-verknüpft. Am Ausgang von Gatter D3.4 steht das codierte Signal zur Verfügung. Über R1 und R2 gelangt es an den Kassettenrecorder.
Der Demodulator ist etwas aufwendiger. Bild 8.5 zeigt das Impulsdiagramm.
Das Eingangssignal wird dem Operationsverstärker N1 zugeführt. An seinem Ausgang liegt das begrenzte Wiedergabesignal, welches durch das D-Flip-Flop D4.1 mit dem Takt T synchronisiert wird. Die synchronisierten Daten werden über den Schalter S1 an den Eingang RxDA der SIO gelegt. Der Schalter S1 legt die Phasenlage der Daten fest (je nach Kassettenrecorder). Bei falscher Phasenlage werden invertierte Daten gelesen. Die Stellung des Schalters muß man bei der Inbetriebnahme durch Versuch ermitteln. Den Empfangstakt RxCA erzeugen die Schaltkreise D4.2, D5 und D6. Der Zähler D6 und das D-Flip-Flop D4.2 bilden einen monostabilen Multivibrator. Die Gatter D5.1 ... D5.4 erzeugen bei jedem Nulldurchgang des Eingangssignals einen kurzen L-Impuls. Dieser Impuls triggert den monostabilen Multivibrator. An Hand von Bild 8.6 soll die Funktionsweise erläutert werden. Am Ausgang Q des Flip-Flop D4.2 sei L-Pegel. Dieser Eingang ist mit dem Ladeeingang des Zählers D6 verbunden. Der an den Dateneingängen (mit den DIL-Schaltern S2 ... S5) vorgewählte Wert wird in den Zähler übernommen. Der Nulldurchgangsimpuls setzt das Flip-Flop D4.2, Ausgang Q wird H. Mit dem nächsten Taktim-

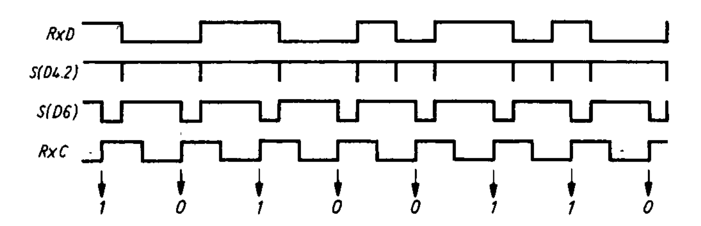


Bild 8.5 Impulsdiagramm des Demodulators

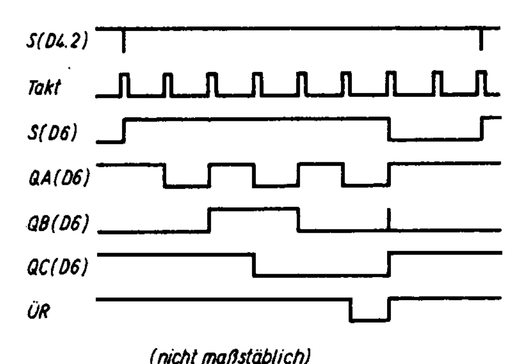


Bild 8.6 Impulsdiagramm des Monoflop

puls T beginnt der Zähler rückwärts zu zählen. Bei Erreichen des Zählerstandes „0" entsteht ein Übertragsimpuls, der den Ausgang Q des Flip-Flop D4.2 auf L schaltet. Somit ist die Ausgangsstellung wieder erreicht. Am Ausgang Q des Flip-Flop entsteht die schon in Bild 8.3 dargestellte Impulsfolge (5). Den Empfangstakt RxCA für die SIO entnimmt man QC des Zählers D6. Mit jeder steigenden Flanke des Empfangstaktes übernimmt die SIO die an RxDA liegende Information in das Empfangsschieberegister. Die Lage dieser Flanke und damit die Impulsdauer des Monoflop werden mit den DIL-Schaltern S2 ... S5 auf 3/4 der Bitlänge eingestellt. Im Mustergerät lädt sich der Zähler mit dem Wert 5.
Die Inbetriebnahme ist mit der in Abschnitt 9. beschriebenen Software nicht schwierig. Es wird ein kurzer Datenblock aufgezeichnet. Werden die Daten bei Wiedergabe nicht gelesen, muß man die Stellung von Schalter S1 (Phasenlage) ändern. Der optimale Abtastzeitpunkt läßt sich dann mit S2 ... S5 einstellen (Richtwert: 5). Die Übertragungsgeschwindigkeit kann durch Verändern der Zeitkonstante des CTC-Kanals 2 den eigenen Wünschen angepaßt werden.
Tabelle 8.1 zeigt den Zusammenhang zwischen der zu programmierenden Zeitkonstante und der Baudrate bei Systemtaktfrequenzen von 2,5 und 2,4596 MHz. Die maximal mögliche Übertragungsrate hängt vom verwendeten Magnetbandgerät ab. Die Übertragungsgeschwindigkeit sollte nicht wesentlich kleiner als 1200 baud sein. Die im Mustergerät gewählte Baudrate von 4800 baud stellt einen guten Kompromiß zwischen möglichst geringer

Tabelle 8.1. Zusammenhang: CTC-Zeitkonstante-Baudrate

TC	Baudrate bei $f_c = 2{,}5$ MHz	bei $f_c = 2{,}4576$ MHz
1	19531	19200
2	9766	9600
3	6510	6400
4	4883	4800
5	3906	3840
6	3255	3200
8	2441	2400
12	1628	1600
16	1220	1200
32	610	600

Fehlerzahl und möglichst hoher Geschwindigkeit dar (1 kByte wird in 2,25 s aufgezeichnet). Hochwertiges Bandmaterial ist Grundvoraussetzung für die erreichten Werte.
Wie bereits erwähnt, werden eventuell auftretende Lesefehler selbständig korrigiert. Ermöglicht wird das durch eine geeignete Formatierung der Daten. Wie man die Daten formatiert, ist in Bild 8.7 dargestellt. Die aufzuzeichnenden Daten werden in einzelne Blöcke zu je 128 Byte (oder weniger, falls die Anzahl der zu übertragenden Byte < 128 beträgt) aufgeteilt. Ein Datenblock beginnt mit 6 Synchronzeichen. Es folgt das Zeichen GS (1 DH), das den Datenblockbeginn kennzeichnet. Das nächste Byte enthält die Anzahl der folgenden Datenbyte. Nach der Übertragung der n Datenbyte folgt das Prüfsummenbyte (die Übertragung von nur einem Prüfsummenbyte hat sich als ausreichend erwiesen).
Zu Beginn jeder Datei wird ein Block übertragen, der alle wichtigen Datenparameter enthält (Kopf). Der Kopf beginnt mit 20 Synchronzeichen. NL (01EH) kennzeichnet den Beginn der Aufzeichnung, wird also nur in Datei 1 (Bild 8.7) übertragen. Die Kennung des Kopfes ist das Zeichen SOH (01H). Die nächsten 8 Zeichen enthalten den Dateinamen. Die weiteren Parameter sind:
– TT Dateityp (maximal 2 Zeichen), zum Beispiel:
'P' Maschinenprogramm mit Autostart
'BA' BASIC-Programme
'A' Textdatei
'BI' Sonstige

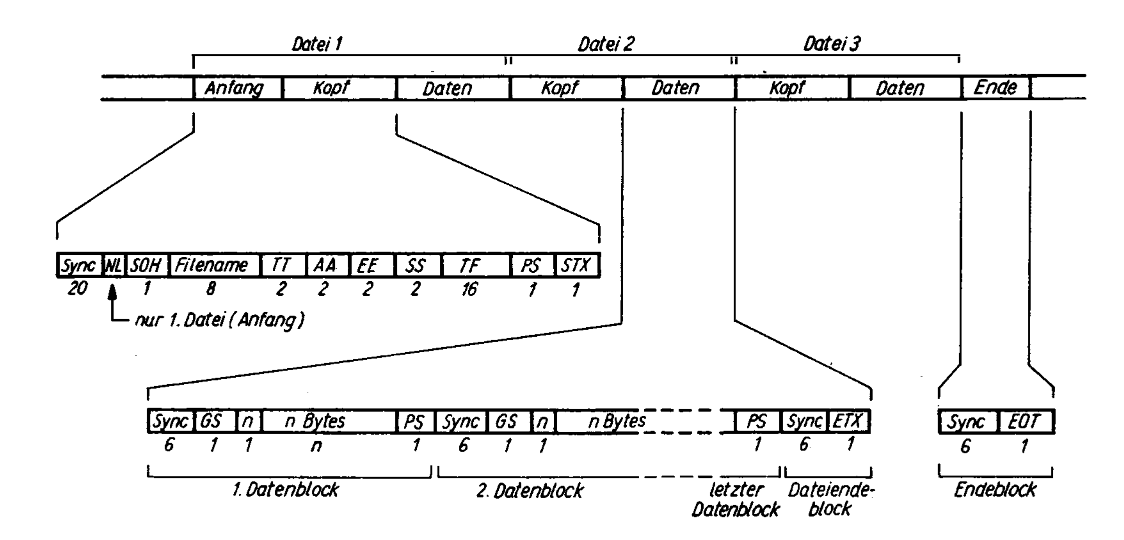


Bild 8.7 Verwendetes Dateiformat für Kassettenaufzeichnung

- AA Anfangsadresse (2 Byte)
- EE Endadresse (2 Byte)
- SS Startadresse für Autostart
- TF Textfeld (16 Zeichen)

Im Textfeld kann beispielsweise das Erstellungsdatum oder der Bearbeitungszustand eingetragen werden. Anschließend folgt die Prüfsumme des Kopfes. STX (O2H) beendet den Kopfblock, zugleich zeigt STX den Beginn der Datenblöcke an. Sind alle Datenblöcke übertragen, erscheinen 6 Synchronzeichen und ETX (03H). Dieser kurze Block beendet die Datei. Die gesamte Datei wird danach noch 2mal übertragen. Die Aufzeichnung endet mit dem Endeblock, der aus 6 Synchronzeichen und dem Zeichen EOT (04H) besteht.

Nicht nur die Hardware, auch die Software ist für die Wiedergabe umfangreicher als für den Aufzeichnungsvorgang. Eine Datei wird eingelesen, wenn der Dateiname der Aufzeichnung mit dem eingegebenen Dateinamen übereinstimmt. Dabei sind alle 8 Zeichen des Namens signifikant. Bei der Wiedergabe werden die einzelnen Datenblöcke mitgezählt. Traten keine Lesefehler auf, so ist nach Erreichen des Dateiendeblocks (ETX) die Wiedergabe beendet. Lagen jedoch Lesefehler vor, so wird die nächste Datei gelesen und alle mit fehlerhafter Prüfsumme „gemerkten" Datenblöcke werden neu geladen. Bei mangelhaftem Bandmaterial oder starken Gleichlaufschwankungen des Kassettenrecorders kann die Anzahl der Lesefehler so groß werden, daß eine Korrektur durch Neuladen der fehlerhaften Datenblöcke nicht mehr möglich ist. In diesem Fall lädt man die gesamte Datei neu. Da jede Datei 3mal aufgezeichnet wurde, ist in der Regel (auch bei weiteren Fehlern) das Laden (und Korrigieren) der Datei möglich. Wird beim Lesen der EOT-Block erreicht, so war keine fehlerfreie Wiedergabe möglich.

Die Software (Abschnitt 9.) organisiert den Datenverkehr. Details sind der Softwarebeschreibung und dem Listing zu entnehmen.

9. Das Monitorprogramm

Bisher wurde vor allen Dingen die Computerhardware beschrieben. Damit der Computer sinnvolle Funktionen ausführen kann, muß ihm ein geeignetes Programm eingegeben werden. Das in diesem Abschnitt beschriebene Monitorprogramm soll den Computer „zum Leben erwecken".
Das Monitorprogramm benötigt als minimale Hardwarevoraussetzung die CPU-Karte (mit 3-kByte-ROM und 1-kByte-RAM), die Tastatur und die Bildschirmsteuerung. Wenn die I/O-Karte eingesetzt wird, sind alle vorhandenen Monitorkommandos ausführbar.
Das Monitorprogramm wurde im wesentlichen unter folgenden Gesichtspunkten geschrieben:

- einfache Bedienung, Fehlertoleranz;
- einfache Ergänzung durch zusätzliche Programmmodule;
- einfache Arbeit mit externen Speichermedien;
- Schnittstellen zu höheren Programmiersprachen (Assembler, BASIC).

Der Monitor wird als ausführliches Assemblerlisting veröffentlicht. Dadurch lernt der Anwender zusätzlich die Funktionsweise kennen, er kann Änderungen einbauen, (zum Beispiel andere Adressen) oder einzelne Teile des Programms anderweitig verwenden.

9.1. Beschreibung der Monitorkommandos

Das Monitorprogramm erwartet nach der Ausgabe des Promptzeichens (&) die Eingabe einer Kommandozeile. Eine Kommandozeile besteht aus dem Kommando selbst und den dazugehörigen Angaben. Die einzelnen Angaben müssen durch ein Komma oder mindestens ein Leerzeichen getrennt werden. Die Eingabe wird durch RETURN (CR) abgeschlossen, der Computer führt nun das eingegebene Kommando aus.
Im folgenden sind die einzelnen Kommandos kurz beschrieben. Den Kommandonamen braucht man nicht auszuschreiben. So kann beispielsweise an Stelle von MEMORY auch MEMOR,

MEMO, MEM, ME oder M geschrieben werden. Alle angegebenen Zeichen sind signifikant. Beginnen verschiedene Kommandonamen mit dem gleichen Buchstaben und ist nur ein Buchstabe angegeben, so wird das zuerst gefundene Kommando angesprungen. M entspricht MEMORY, aber nicht MOVE, für MOVE muß mindestens MO eingegeben werden. Klarheit über die Reihenfolge der Kommandos schafft das Kommando HELP.

– HELP Anzeige aller im Speicher vorhandenen Kommandos HELP CR

Sowohl die Kommandos des Monitors als auch die vom Anwender definierten werden gelistet. Im gesamten Speicher wird nach dem Kommandorahmen (0EDH, 0FFH, Kommandozeichenkette, 00 H) gesucht, und die Kommandozeichenketten werden ausgegeben (siehe auch Abschnitt 9.2.).

– MEMORY Anzeigen und Modifizieren von Speicherbereichen/
MEMORY [Option] Startadresse CR

Es wird ein Speicherbereich ab Startadresse und 1/4 kByte Länge in hexadezimaler Form und in Textdarstellung ausgegeben. Bei der Textdarstellung erscheinen nicht darstellbare Codes als Punkt. Bei Angabe der Option @ A beginnt jede neue Zeile zusätzlich mit der entsprechenden Adresse. Dabei wird die Textdarstellung unterdrückt.

Mit dem MEMORY-Befehl lassen sich auch Speicherzellen modifizieren. Ein spezieller Cursor zeigt auf die zu modifizierende Speicherzelle. Die Cursorposition und damit die Adresse können mit den 4 Cursortasten (links, rechts, hoch, tief) verändert werden. Die Eingabe erfolgt hexadezimal (Mode H) oder als Textzeichen (Mode A). Zum Umschalten in den jeweils anderen Modus benutzt man die Escape-Taste (ESC).

Eine neue Startadresse läßt sich nach Drücken der Taste Q (Quit) eingeben. Folgt an Stelle einer Adresse ein weiters Q, so wird das Programm MEMORY nach Drücken der RETURN-Taste verlassen.

– MOVE Umladen von Speicherbereichen
MOVE, Startadresse, Endadresse, Zieladresse CR

Es wird der Bereich von Startadresse bis Endadresse nach Zieladresse kopiert. Dabei können sich die Speicherbereiche auch überlappen. Nach Ausführung des MOVE-Befehls wartet das Programm auf eine Tastatureingabe: Q (Quit) oder RETURN (oder irgendeine andere Taste). Im letzteren Fall springt das Programm

nach MEMORY und zeigt den Speicherbereich ab Startadresse an.

- FILL   Füllen eines Speicherbereichs mit einer Konstanten
  FILL, Startadresse, Endadresse [Byte] CR

Das angegebene Byte füllt den entsprechenden Speicherbereich. Fehlt das Füllbyte, so wird das Byte als 00H angenommen (schnelles Löschen eines Speicherbereichs).

- IN, OUT   Bedienen von Ports
  IN, Portadresse CR
  OUT, Portadresse, Datenbyte CR

Mit dem IN-Befehl lassen sich Ports direkt lesen. Das gelesene Byte wird auf dem Consolkanal (Bildschirm) ausgegeben.
Mit dem OUT-Befehl können zum Beispiel Ports initialisiert werden.

Beispiel: OUT 0 CR
  OUT 0 CR entspricht dem Kommando OFF

- OFF   Ausschalten des Computers
  OFF CR

Nach etwa 20 s schaltet sich der Computer ab. Den Abschaltvorgang kann man durch RESET, NMI oder durch die Sequenz IN 0 CR, IN 0 CR abbrechen.

- SAVE   Abspeichern auf Magnetband
  SAVE, Filename, Filetyp, Anfangsadr., Endadr.
  [Eintrittspunkt] [Textfeld] CR

SAVE zeichnet den Speicherbereich von Anfangs- bis Endadresse auf Magnetband auf. Der Name des Programms kann beliebig lang sein, es werden aber nur die ersten 8 Zeichen als Filename verwendet. Durch Angabe eines Filetyps läßt sich die Datei besonders kennzeichnen, beispielsweise können BASIC-Programme durch @ BA oder Textdateien durch @ A gekennzeichnet werden. Bei der Angabe eines Filetyps sind 2 Zeichen signifikant. Ist kein Filetyp angegeben, so wird vom SAVE-Programm automatisch P eingesetzt. P kennzeichnet die Datei als lauffähiges Maschinenprogramm. Dateien vom Typ P erfordern noch einen weiteren Parameter, nämlich den Eintrittspunkt des Programms. Bei nicht angegebenem Eintrittspunkt wird vom SAVE-Programm automatisch der Restart-Eintrittspunkt des Monitors eingetragen. Schließlich können im Textfeld noch zusätzliche Informationen, z. B. das Erstellungsdatum, folgen. Es werden 16 Textzeichen (nach;) aufgezeichnet. Dieses Textfeld werten die hier beschriebenen Monitorkommandos nicht aus. Der Anwender kann den Monitor aber

durch weitere Kommandos ergänzen, die bei Bedarf dieses Textfeld auch auswerten (von den Autoren wurde ein Programm CAT geschrieben, das alle auf Band befindlichen Dateien mit ihren Eigenschaften einschließlich Textfeld auflistet).

– LOAD   Laden von Magnetband
LOAD, Filename [Option] [Offset] CR

Ein mit SAVE aufgezeichneter Speicherbereich wird wieder eingelesen. Wenn der angegebene Filename mit dem aufgezeichneten Namen übereinstimmt, wird die Datei geladen. Man erkennt das an der Aufschrift „FILE FOUND“. Nach dem Laden, falls es sich um ein abarbeitbares Maschinenprogramm (Filetyp P) handelt, startet das Programm automatisch (Eintrittsadresse), es sei denn, es wurde die Option @ Q angegeben. @ Q unterdrückt den Autostart. Bei Bedarf kann noch ein Offsetwert angegeben werden. Der Wert addiert sich zur Ladeadresse und ermöglicht das Laden auf beliebige Speicherbereiche. Dabei ist natürlich der eventuelle Autostart unterdrückt. Nach dem Laden werden die eventuell aufgetretenen Lesefehler angezeigt. Diese Fehler wurden aber vom LOAD-Programm korrigiert. Nur wenn „BAD FILE“ erscheint, war keine Korrektur möglich, d. h., die im RAM befindlichen Daten sind fehlerhaft.

– READ   Lochstreifen lesen
READ [Option] [Offset] CR

Dieser Befehl ermöglicht das Lesen von MC-Lochstreifen (Intel-Hex-Format) vom Lochstreifenlesekanal RI. Wird die Option @ A angegeben, dann stoppt das Programm nach Einlesen 1. Adresse, gibt diese Adresse aus und wartet auf eine Eingabe. Drückt man ohne Eingabe RETURN, so wird, beginnend bei der angezeigten Startadresse, eingelesen. Es kann aber auch eine andere Adresse eingegeben werden. Nach dem Drücken der RETURN-Taste wird dann das Einlesen, beginnend bei der neuen Adresse, fortgesetzt. Eine andere Möglichkeit, den Lochstreifen „verschoben“ einzulesen, besteht darin, daß man einen Offsetwert angibt.
Beim Auftreten eines Lesefehlers stoppt das Programm. Mit Hilfe der Taste Q kann nun das READ-Programm verlassen werden. Es ist aber auch möglich, den fehlerhaften Block noch einmal einzulesen (Lochstreifen zum Blockanfang [:] zurücktransportieren, dann RETURN).

- NMI Vektor eintragen
  NMI CR

Es wird auf die Adresse 066H ein Rücksprung zum Monitor eingetragen. Den nichtmaskierten Interrupt kann man dann als „Notbremse" nutzen. Die CPU-Register werden gerettet. Ein ununterbrochenes Programm wird durch GO PC mit den geretteten Registerwerten wieder gestartet. Dabei ist zu beachten, daß sich nur Programme wieder starten lassen, die nicht den Systemstack verwenden.

- GO Start von Programmen
  GO, Startadresse [Breakpoint] CR
  GO PC CR

Mit GO adr. läßt sich ein Programm starten. Der Stackpointer wird auf die Adresse USTCK (Userstack) gesetzt. Wenn das Anwenderprogramm diesen Stack benutzt, kann es durch RET (0C9H) in den Monitor zurückkehren. Es ist möglich, zusätzlich den Breakpoint zu setzen, wenn das Anwenderprogramm zu Testzwecken unterbrochen werden soll. Der Breakpoint muß auf den Anfang eines Befehls gesetzt werden. Das GO-Programm rettet das dort stehende Byte und trägt 0FFH (Restart 38 H) ein. Bei Erreichen des Breakpoints wird das ursprüngliche Byte wieder zurückgeladen, anschließend werden alle CPU-Register angezeigt. Sie lassen sich nun gegebenenfalls verändern (REG). Das Anwenderprogramm wird durch GO PC fortgesetzt. Bei der Anwendung von GO PC wird der RAM-Bereich der Userregister in die CPU kopiert (siehe REG). Jedes Programm kann statt mit GO adr. auch mit GO PC gestartet werden (SP berücksichtigen!). GO ohne weitere Angaben führt keinen Programmstart aus, sondern eignet sich dazu, einen nicht erreichten Breakpoint zu löschen.

- REG Anzeigen und Modifizieren der CPU-Register
  REG [reg Byte [reg Byte . . .]] CR

Das Kommando REG ohne weitere Angaben zeigt die CPU-Register an. Die Register verändert man durch die Eingabe des Registers und des gewünschten Wertes. Beispielweise bewirkt das Kommando

REG A0, SP1000, PC100, H'0

die Belegung der Register A, H', SP und PC mit den angegebenen Werten. Genaugenommen werden die entsprechenden RAM-Zellen (Userregister UREG) verändert.

– CHECKSUM Prüfsummenberechnung
CHECKSUM, Startadr., Endadr. CR

Dieses Kommando berechnet von dem angegebenen Speicherbereich eine 2-Byte-Prüfsumme.

– = Adreßrechnung
= ,x, y, CR

Diese Eingabe bewirkt folgende Berechnung und Ausgabe:
SUM : x + y DIF : x – y REL: **
Dabei bedeutet REL die relative Distanz bei Sprungbefehlen (JR, DJNZ usw.). Ist die Distanz zu groß, so werden an Stelle eines Byte 2 Sterne ausgegeben.

– FIND Suchen von Bytes oder Zeichenketten
FIND, Startadr., Endadr., Byte 1, Byte 2 . . . Byte n
CR FIND, A, Startadr., Endadr., Zeichenkette CR

FIND ermöglicht das Suchen nach einer Bytefolge oder nach einer Zeichenkette in dem Speicherbereich von Start- bis Endadresse. Auf dem Bildschirm werden alle Adressen ausgegeben, auf denen die Zeichenkette oder Bytefolge gefunden wurden.

– SIZE Speichertest
SIZE CR

SIZE führt einen einfachen, nicht zerstörenden RAM-Test durch, gibt den höchsten als RAM verfügbaren Speicherplatz aus und erwartet eine Eingabe. Die ermittelte Adresse oder eine jetzt einzugebende Adresse wird nach RETURN in die Zelle MSIZE geladen. Der Monitor selbst benötigt die Angabe über den höchsten Speicherplatz nicht. Aber verschiedene Programme (z. B. BASIC-Interpreter) brauchen in der Initialisierungsphase die Angabe über den höchsten zur Verfügung stehenden Speicherplatz, der entweder das RAM-Ende oder eine andere Adresse sein kann.

– EOF Dateiendezeichen definieren
EOF CR

Verschiedene Programme (BASIC-Interpreter, Assembler usw.) benötigen beim Einlesen von Daten und Programmen ein End-of-File-Zeichen. Fehlt dieses Zeichen auf dem Datenträger (z. B. Lochstreifen), so läßt sich die Lesefunktion dieser Programme nicht beenden. Um dieses Problem zu lösen, wurde der Monitor so ausgelegt, daß nach Drücken von CTRL C EOF-Zeichen gesendet werden. Da die verschiedenen Programme unterschiedliche EOF-Zeichen voraussetzen, wurde dieses Kommando geschaffen. Es

zeigt das derzeitig gültige EOF-Zeichen an. Das Zeichen kann wieder übernommen oder durch Neueingabe verändert werden.

– ASN Gerätezuweisung
ASNRI = RI CR
ASNRI = CRI CR
ASNRI = USR, Adresse CR

Mit diesem Kommando kann man die jeweilige Gerätezuweisung verändern (siehe auch Abschnitt 9.2.). Hat beispielsweise ein Anwenderprogramm nur eine Lochstreifenschnittstelle, so lassen sich die Daten bei der Zuweisung RI = RI vom Lochstreifenleser oder bei RI = CRI vom Kassettenmagnetbandgerät einlesen.
Im vorliegenden Monitor sind nur wenige Zuweisungen fest vorhanden. Eine Erweiterung ist leicht möglich (siehe Abschnitt 9.2.). Man muß allerdings beim Verändern der Consolzuweisung aufpassen, sonst wird das Bedienen des Computers in Frage gestellt (Die Gerätezuweisungen wirken auch auf den Monitor!). In solchen Fällen hilft dann nur noch RESET.

9.2. Schnittstellen und Erweiterbarkeit

Der Monitor hat ab Adresse 0F000H einige Eintrittspunkte, die im folgenden kurz erläutert werden:

JMP BEGIN Neustart des Monitors
JMP CI Consoleingabe (Tastatur, Zeichen in A)
JMP RI Lesekanal (Lochstreifenleser, Zeichen in A)
JMP COE Consolausgabe (Bildschirm, Zeichen C)
JMP POE Punchkanal (Lochstreifenstanzer, Zeichen in C)
JMP LOE Listkanal (Drucker, Zeichen in C)
JMP CSTS Consolstatus (A = 0 keine Taste, A = 0FFH gedrückt)
JMP CRI Kassettenlesekanal (Zeichen in A)
JMP CPOE Kassettenpunchkanal (Zeichen in C)
JMP MEMSI höchster Speicherplatz (A = Low, B = High-Teil der Adresse)
JMP MAIN Restart des Monitors
JMP EXT Aufruf von Monitorunterprogrammen (C = Nr. des Programms entsprechend ETAB)
DA FINA Beginn RAM-Bereich des Monitors.

Ein Teil dieser Sprünge führt in den RAM-Bereich des Monitors.

Von dort wird dann zur eigentlichen Geräteroutine gesprungen. Dadurch ist eine einfache Gerätezuweisung möglich. Beim Neustart benutzt man folgende Zuordnung (ATAB):

Consoleingabe	CI	—— CI	Tastatur
Lesekanal	RI	—— RI	Lochstreifenleser
Consolausgabe	COE	—— CO	Bildschirm
Punchkanal	POE	—— CPO	Kassettenausgabe
Listkanal	LOE	—— CO	Bildschirm
Kassettenlesekanal	CRI	—— CRI	Kassetteneingabe
Kassettenpunchkanal	CPO	—— CPO	Kassettenausgabe

Bei Veränderung durch ASN wird auf den entsprechenden RAM-Zellen nur die jeweilige Gerätetreiberadresse eingetragen. Weitere Einsprungstellen (wenn Zeichen im Akkumulator übergeben werden sollen) sind:

FINA + 0B6H Listkanal
FINA + 0BAH Punchkanal
FINA + 0BEH Kassettenpunchkanal
FINA + 0C2H Consolausgabekanal

Wie man die Schnittstelle benutzt und den Monitor erweitert, belegen 3 einfache Beispiele.

Im 1. Fall soll eine Schreibmaschinenfunktion realisiert werden. Der Aufruf erfolgt durch Eingabe Typ und RETURN

```
         DB 0EDH        ; Kennzeichnungsrahmen
         DB 0FFH        ; wird von MAIN gesucht
         DB 'TYP'       ; Kommandoname
         DB 00H         ; Ende des Rahmens
LOOP:    CALL 0F003H    ; Zeichen von Tastatur holen
         CMP 03H        ; CTRL C gedrückt?
         Rz             ; Ja, zum Monitor zurück
         LD C,A
         CALL 0F009H    ; Zeichen ausgeben
         JR LOOP-#
```

Der Monitor (Routine MAIN) sucht den Kennzeichnungsrahmen. Wenn die Bytefolge 0EDH und 0FFH gefunden wurde, wird die folgende Zeichenkette mit der im Textpuffer abgelegten Kommandozeichenkette verglichen. Sind beide Zeichenketten identisch und folgt im Speicher das Byte 00H, so wird die dem Rahmenendebyte folgende Adresse von MAIN aus angesprungen. In diesem Beispiel handelt es sich um die Marke LOOP. Dabei ist es gleichgültig, wo die Folge 0EDH, 0FFH, Zeichenkette, 00H im Speicher

steht. Der gesamte Speicher wird nach dem eingegebenen Kommando durchsucht. Die Routine MAIN beginnt die Suche ab Marke START – 1000H (0E000H). Das bedeutet, daß beispielsweise eine veränderte MEMORY-Routine ab Adresse 0E000H abgelegt und bei Eingabe des Befehls MEMORY aufgerufen werden könnte. Die im Monitor-EPROM enthaltene Routine MEMORY wird in diesem Fall nicht mehr gefunden. Jedes Anwenderprogramm kann man mit dem (im Listing HEAD genannten) Rahmen, dem Namen und anschließenden Byte 00H versehen. Auf diese einfache Art und Weise ist es möglich, jedes Programm mit seinem Namen vom Monitor aus aufzurufen. Die Rückkehr zum Monitor erfolgt durch einen RET-Befehl.

Im 2. Beispiel sollen alle Zeichen, die über den Listkanal an den Monitor gelangen, auf Magnetband aufgezeichnet werden. Benötigt wird dazu eine neue Gerätezuweisung LO=CPO.

```
DB 0EDH
DB 0FFH
DB 'ASNLO=CPO'
DB 00H
LD HL, CPO1                    ; Adresse Kassettentreiber
LD (LO+1),HL                   ; ins RAM laden
RET                            ; zurück zum Monitor
```

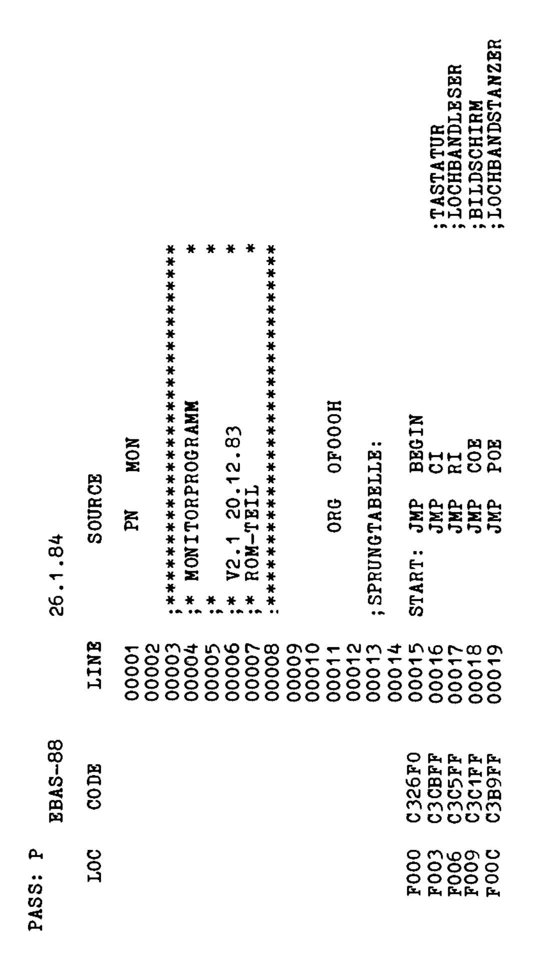
Im 3. Beispiel soll der Hexadezimalcode eines von der Tastatur eingegebenen Zeichens auf dem Bildschirm ausgegeben werden.

```
ENTRY:    CALL 0F003H   ; Zeichen von Tastatur holen
          PUSH AF       ; retten
          LD C, A
          CALL 0F009H   ; Zeichen ausgeben
          LD C, 5       ; Code für PRINT
          CALL 0F021H   ; EXT anspringen
          DB ' = '      ; Zeichenkette ausgeben
          DB 00H        ; bis 0
          POP AF        ; Zeichen zurück
          LD C, 3       ; Code für HEX-Ausgabe
          CALL 0F021H   ; EXT anspringen
          RET           ; zurück zum Monitor
```

Dieses kurze Programm zeigt, wie man den Programmteil EXT anwendet. Die Codes enthält Tabelle ETAB (siehe Listing). Auch die Funktionsweise der einzelnen Unterprogramme ist dem Listing zu entnehmen.

Die 3 Beispielprogramme sollten die Anwendung der Monitorroutinen demonstrieren. Weitere Einzelheiten findet man im ausführlichen Assemblerlisting

9.3. Das Programm



```
EBAS-88: R

PASS: P        EBAS-88          26.1.84

LOC   CODE     LINE   SOURCE

               00001         PN    MON
               00002
               00003  ;*************************
               00004  ;* MONITORPROGRAMM       *
               00005  ;*                       *
               00006  ;* V2.1 20.12.83         *
               00007  ;* ROM-TEIL              *
               00008  ;*************************
               00009
               00010
               00011         ORG   0F000H
               00012
               00013  ;SPRUNGTABELLE:
               00014
F000  C326F0   00015  START: JMP   BEGIN
F003  C3CBFF   00016         JMP   CI         ;TASTATUR
F006  C3C5FF   00017         JMP   RI         ;LOCHBANDLESER
F009  C3C1FF   00018         JMP   COE        ;BILDSCHIRM
F00C  C3B9FF   00019         JMP   POE        ;LOCHBANDSTANZER
```

```
F00F  C3B5FF   00020         JMP  LOE                   ;DRUCKER
F012  C3C2FA   00021         JMP  CSTS                  ;KONSOLSTATUS
F015  C3C8FF   00022         JMP  CRI                   ;CASSETTENEINGABE
F018  C3BDFF   00023         JMP  CPOE                  ;CASSETTENAUSGABE
F01B  C32AFA   00024         JMP  MEMSI                 ;SPEICHERBEREICH
F01E  C36CF0   00025  RSTAR: JMP  MAIN                  ;RESTART MONITOR
F021  C34EFA   00026         JMP  EXT                   ;MONITOR FUNKTIONEN
F024  00FF     00027         DA   FINA                  ;FUER EXTERN
               00028
               00029
F026  31FEFE   00030  BEGIN: LD   SP,STACK
F029  2189FB   00031         LD   HL,ATAB               ;GERAETEZUWEISUNG
F02C  11B1FF   00032         LD   DE,MSIZE              ;INITIALISIEREN
F02F  012300   00033         LD   BC,35                 ;SPRUNGTABELLE INS RAM LADEN
F032  EDB0     00034         LDIR
F034  CD74F1   00035         CALL PRINT                 ;MELDUNG AUSGEBEN
F037  0C       00036         DB   CLS
F038           00037         DB   ' H.MON V2.1 12/83'
F049  1E       00038         DB   NL
F04A  00       00039         DB   0
F04B  CD61FA   00040         CALL CINIT                 ;KASSETTENPORT INIT
F04E  CD7BFA   00041         CALL RINIT                 ;LESERPORT INIT
F051  CD32FA   00042         CALL MEMCH                 ;RAMTEST
F054  22B1FF   00043         LD   (MSIZE),HL
F057  1813     00044         JR   MAIN-#
               00045
               00046
               00047  ;******************************
               00048  ;* MAIN                        *
               00049  ;******************************
```

```
                    00050
F059  CD74F1        00051  ERROR: CALL PRINT
F05C                00052         DB   'ILLEGAL COMMAND'
F06B  00            00053         DB   0
                    00054
                    00055  ;BEFEHLSZEILE EINGEBEN UND PROGRAMM ANSPRINGEN
                    00056
F06C  31FEFE        00057  MAIN:  LD   SP,STACK          ;NEU LADEN
F06F  CD81F1        00058         CALL INL               ;EINGABE BEFEHLSZEILE
F072  CD55F1        00059         CALL FNEXT             ;SPACE AUSBLENDEN
F075  38F5          00060         JRC  MAIN-#            ;NUR BEI CR
F077  ED5340FF      00061         LD   (TXTPT),DE
F07B  2100E0        00062         LD   HL,START-1000H    ;SUCHBEGINN
F07E  01FFFF        00063         LD   BC,0FFFFH         ;64K DURCHSUCHEN
F081  3EED          00064  MAIN1: LD   A,L(HEAD)
F083  EDB1          00065         CPIR                   ;NACH ED SUCHEN
F085  20D2          00066         JRNZ ERROR-#           ;FEHLER
F087  3EFF          00067         LD   A,H(HEAD)         ;FOLGT FF?
F089  BE            00068         CMP  (HL)
F08A  20F5          00069         JRNZ MAIN1-#           ;NEIN
F08C  ED5B40FF      00070         LD   DE,(TXTPT)
F090  1A            00071  MAIN2: LD   A,(DE)            ;VERGLEICHEN
F091  23            00072         INC  HL
F092  13            00073         INC  DE
F093  0B            00074         DEC  BC
F094  BE            00075         CMP  (HL)
F095  28F9          00076         JRZ  MAIN2-#           ;GLEICH
F097  FE20          00077         CMP  ' '               ;TEST TRENNZEICHEN/CR
F099  2808          00078         JRZ  MAIN3-#
F09B  FE2C          00079         CMP  ','
```

```
F09D  2804     00080         JRZ  MAIN3-#
F09F  FE0D     00081         CMP  CR
F0A1  20DE     00082         JRNZ MAIN1-#              ;WEITERSUCHEN
               00083  ;EVT.DOCH KEIN BEFEHL?
F0A3  AF       00084  MAIN3: XOR  A                    ;TEST RAHMEN
F0A4  2B       00085         DEC  HL
F0A5  C5       00086         PUSH BC
F0A6  010800   00087         LD   BC,8
F0A9  EDB1     00088         CPIR
F0AB  C1       00089         POP  BC
F0AC  20D3     00090         JRNZ MAIN1-#
               00091  ;HL ZEIGT JETZT AUF 1.BEFEHL
               00092  ;DE ZEIGT JETZT AUF 1.ARGUMENT
F0AE  016CF0   00093         LD   BC,MAIN              ;RUECKSPRUNG
F0B1  C5       00094         PUSH BC
F0B2  E9       00095         JMP  (HL)                 ;SPRUNG ZUM PROGRAMM
               00096
               00097
               00098  ;EINZELNE ARGUMENTE HOLEN
               00099
F0B3  CD55F1   00100  GFINA: CALL FNEXT                ;FILENAME
F0B6  2100FF   00101         LD   HL,FINA
F0B9  0608     00102         LD   B,8
F0BB  CD55F1   00103  GFIN1: CALL FNEXT
F0BE  280C     00104         JRZ  GFIN2-#              ;WEIL TRENNZEICHEN
F0C0  77       00105         LD   (HL),A
F0C1  23       00106         INC  HL
F0C2  13       00107         INC  DE
F0C3  10F6     00108         DJNZ GFIN1-#
F0C5  CD55F1   00109  GFIN3: CALL FNEXT                ;UEBERFLUESSIGE ZEICHEN IGNORIEREN
```

```
F0C8  C8        00110         RZ
F0C9  13        00111         INC  DE
F0CA  18F9      00112         JR   GFIN3-#
F0CC  3620      00113 GFIN2:  LD   (HL),' '           ;MIT SPACE AUFFUELLEN
F0CE  23        00114         INC  HL
F0CF  10FB      00115         DJNZ GFIN2-#
F0D1  C9        00116 NICHT:  RET
                00117
F0D2  CD55F1    00118 GFITY:  CALL FNEXT              ;FILETYP/OPTION
F0D5  2108FF    00119         LD   HL,FITY
F0D8  0602      00120         LD   B,2
F0DA  FE40      00121         CMP  '@'
F0DC  20EE      00122         JRNZ GFIN2-#
F0DE  13        00123         INC  DE
F0DF  18DA      00124         JR   GFIN1-#
                00125
F0E1  CD55F1    00126 GCOM:   CALL FNEXT              ;ZUSAETZL.ANGABEN
F0E4  2110FF    00127         LD   HL,DATA
F0E7  0610      00128         LD   B,16
F0E9  FE3B      00129         CMP  ';'
F0EB  20DF      00130         JRNZ GFIN2-#
F0ED  13        00131 GCOM1:  INC  DE
F0EE  1A        00132         LD   A,(DE)
F0EF  FE0D      00133         CMP  CR
F0F1  28D9      00134         JRZ  GFIN2-#
F0F3  77        00135         LD   (HL),A
F0F4  23        00136         INC  HL
F0F5  10F6      00137         DJNZ GCOM1-#
F0F7  C9        00138         RET
                00139
```

```
                00140  ;ASCII -> HEX
                00141
F0F8  3E20      00142  INHEX: LD   A,' '          ;HEX VON TASTATUR HOLEN
F0FA  CDC2FF    00143         CALL CO
F0FD  CD87F1    00144         CALL INL00
                00145
F100  CD55F1    00146  ASHEX: CALL FNEXT          ;KONVERTIEREN
F103  CD20F1    00147         CALL CNVBN
F106  1A        00148         LD   A,(DE)
F107  D8        00149         RC                  ;KEIN HEX
F108  C5        00150         PUSH BC
F109  210000    00151         LD   HL,0
F10C  1A        00152  ASH1:  LD   A,(DE)
F10D  CD20F1    00153         CALL CNVBN
F110  380B      00154         JRC  ASH4-#
F112  29        00155  ASH3:  ADD  HL,HL          ;SCHIEBEN
F113  29        00156         ADD  HL,HL
F114  29        00157         ADD  HL,HL
F115  29        00158         ADD  HL,HL
F116  0600      00159         LD   B,0
F118  4F        00160         LD   C,A
F119  09        00161         ADD  HL,BC
F11A  13        00162         INC  DE
F11B  18EF      00163         JR   ASH1-#
F11D  C1        00164  ASH4:  POP  BC
F11E  AF        00165         XOR  A
F11F  C9        00166         RET
                00167
F120  D630      00168  CNVBN: SUB  30H
F122  D8        00169         RC
```

```
F123  FE0A     00170          CMP  10
F125  3F       00171          CCF
F126  D0       00172          RNC
F127  FE11     00173          CMP  11H
F129  D8       00174          RC
F12A  FE17     00175          CMP  17H
F12C  3F       00176          CCF
F12D  D8       00177          RC
F12E  D607     00178          SUB  7
F130  C9       00179          RET
               00180
               00181
               00182  ;OUTHEX
               00183  ;WANDELT HEX=>ASCII
               00184  ;OUTH GIBT BYTE <A> AUS
               00185
F131  F5       00186  OUTH:   PUSH AF
F132  1F       00187          RRA                        ;SCHIEBEN
F133  1F       00188          RRA
F134  1F       00189          RRA
F135  1F       00190          RRA
F136  CD3AF1   00191          CALL OUTH1
F139  F1       00192          POP  AF
F13A  F5       00193  OUTH1:  PUSH AF
F13B  E60F     00194          AND  00001111B
F13D  C630     00195          ADD  030H
F13F  FE3A     00196          CMP  03AH
F141  3802     00197          JRC  OUTH2-#
F143  C607     00198          ADD  07
F145  CDC2FF   00199  OUTH2:  CALL CO
```

```
F148  F1        00200         POP   AF
F149  C9        00201         RET
                00202
                00203  ;OUTHL GIBT WORT <HL> AUS
                00204
F14A  F5        00205  OUTHL: PUSH  AF
F14B  7C        00206         LD    A,H
F14C  CD31F1    00207         CALL  OUTH
F14F  7D        00208         LD    A,L
F150  CD31F1    00209         CALL  OUTH
F153  F1        00210         POP   AF
F154  C9        00211         RET
                00212
F155  1A        00213  FNEXT: LD    A,(DE)
F156  FE20      00214         CMP   ' '
F158  200D      00215         JRNZ  FNEX3-#
F15A  13        00216  FNEX1: INC   DE
F15B  1A        00217         LD    A,(DE)
F15C  FE20      00218         CMP   ' '
F15E  28FA      00219         JRZ   FNEX1-#              ;SPACE AUSBLENDEN
F160  FE0D      00220         CMP   CR
F162  2805      00221         JRZ   FNEX4-#
F164  AF        00222  FNEX2: XOR   A
F165  1A        00223         LD    A,(DE)
F166  C9        00224         RET
F167  FE0D      00225  FNEX3: CMP   CR
F169  37        00226  FNEX4: SCF
F16A  C8        00227         RZ
F16B  3F        00228         CCF
F16C  FE2C      00229         CMP   ','
```

```
F16E  13        00230          INC  DE
F16F  28F3      00231          JRZ  FNEX2-#
F171  B7        00232          OR   A
F172  1B        00233          DEC  DE
F173  C9        00234          RET
                00235
                00236  ;PRINT GIBT ZEICHENKETTE BIS 0 AUS
                00237
F174  E3        00238  PRINT:  EX   (SP),HL
F175  7E        00239  PR1:    LD   A,(HL)
F176  23        00240          INC  HL
F177  B7        00241          OR   A                    ;PRINT-ENDE?
F178  2805      00242          JRZ  PR2-#                ;JA
F17A  CDC2FF    00243          CALL CO
F17D  18F6      00244          JR   PR1-#
F17F  E3        00245  PR2:    EX   (SP),HL
F180  C9        00246          RET
                00247
                00248  ;BEFEHLSZEILEN-EINGABE
                00249
F181  CD74F1    00250  INL:    CALL PRINT
F184  1E        00251          DB   NL
F185  26        00252          DB   26H                  ;PROMT
F186  00        00253          DB   0
F187  1142FF    00254  INL00:  LD   DE,TXTBU
F18A  0651      00255          LD   B,BULEN
F18C  CDCBFF    00256  INL1:   CALL CI
F18F  FE06      00257          CMP  ERASE                ;ZEILE LOESCHEN
F191  28EE      00258          JRZ  INL-#
F193  FE08      00259          CMP  CURL                 ;BACKSTEP
```

```
F195  280F     00260         JRZ  INL3-#
F197  FE0D     00261         CMP  CR                  ;RETURN TASTE?
F199  281B     00262         JRZ  INL2-#
F19B  FE20     00263         CMP  20H
F19D  38ED     00264         JRC  INL1-#              ;NICHT DARSTELLBARES ZEICHEN
F19F  12       00265         LD   (DE),A
F1A0  CDC2FF   00266         CALL CO
F1A3  13       00267         INC  DE
F1A4  10E6     00268         DJNZ INL1-#              ;WEITER
F1A6  78       00269  INL3:  LD   A,B                 ;BACKSTEP
F1A7  FE51     00270         CMP  BULEN
F1A9  28E1     00271         JRZ  INL1-#
F1AB  1B       00272         DEC  DE
F1AC  04       00273         INC  B
F1AD  CD74F1   00274         CALL PRINT
F1B0  08       00275         DB   CURL
F1B1           00276         DB   ' '
F1B2  08       00277         DB   CURL
F1B3  00       00278         DB   0
F1B4  18D6     00279         JR   INL1-#
F1B6  12       00280  INL2:  LD   (DE),A              ;CR
F1B7  3E1E     00281         LD   A,NL                ;NEUE ZEILE
F1B9  CDC2FF   00282         CALL CO
F1BC  1142FF   00283         LD   DE,TXTBU            ;ADR. TEXTPUFFER
F1BF  C9       00284         RET
               00285
               00286
               00287  ;*****************************
               00288  ;* MEMORY                    *
               00289  ;*****************************
```

```
                  00290
                  00291  ;ALLE BEFEHLE MIT ! ENTFALLEN BZW ZUSAETZLICH BEI 80*24 CRT
                  00292
F1C0  EDFF        00293          DA    HEAD
F1C2              00294          DB    'MEMORY'
F1C8  00          00295          NOP
F1C9  CDD2F0      00296          CALL  GFITY                  ;!
F1CC  CD00F1      00297          CALL  ASHEX
F1CF  3803        00298  MEMO1:  JRC   MEM-#
F1D1  220AFF      00299          LD    (ARG1),HL
F1D4  FE51        00300  MEM:    CMP   'Q'
F1D6  C8          00301          RZ
F1D7  3E0C        00302          LD    A,CLS                  ;BILDSCHIRM LOESCHEN
F1D9  CDC2FF      00303          CALL  CO
F1DC  3A08FF      00304          LD    A,(FITY)               ;!
F1DF  FE41        00305          CMP   'A'                    ;!
F1E1  3E48        00306          LD    A,'H'                  ;MODE HEX
F1E3  2002        00307          JRNZ  MEMOO-#                ;!
F1E5  F680        00308          OR    80H                    ;!
F1E7  3210FF      00309  MEMOO:  LD    (MEMMO),A
F1EA  2A0AFF      00310  MEMO:   LD    HL,(ARG1)
F1ED  220CFF      00311          LD    (ARG2),HL              ;ARG2=ARG1
                  00312  ;AUSGABE
F1F0  3E0E        00313  MEM1:   LD    A,HOME
F1F2  CDC2FF      00314          CALL  CO                     ;CRT HOME
F1F5  3A10FF      00315          LD    A,(MEMMO)              ;MODE AUSGEBEN
F1F8  CBBF        00316          RES   7,A                    ;!
F1FA  CDC2FF      00317          CALL  CO
F1FD  CD74F1      00318          CALL  PRINT
F200              00319          DB    '  CURSOR:'
```

```
F209  00      00320         DB   0
F20A  2A0CFF  00321  MEM2:  LD   HL,(ARG2)
F20D  7D      00322         LD   A,L              ;MODULO RECHNEN
F20E  E6F0    00323         AND  0-MEMB
F210  6F      00324         LD   L,A
F211  220CFF  00325         LD   (ARG2),HL        ;ADR 1.BYTE
F214  EB      00326         EX   DE,HL            ;DE=BEGINN
F215  2A0AFF  00327         LD   HL,(ARG1)        ;CURSOR
F218  CD4AF1  00328         CALL OUTHL            ;AUSGEBEN
F21B  3E1E    00329         LD   A,NL
F21D  CDC2FF  00330         CALL CO
F220  ED52    00331         SBC  HL,DE
F222  2C      00332         INC  L                ;DISTANZ ZUM CURSOR
F223  3A10FF  00333         LD   A,(MEMMO)
F226  67      00334         LD   H,A              ;MODE
F227  0610    00335         LD   B,MEML           ;ZEILENANZAHL
F229  C5      00336  MEM3:  PUSH BC               ;RETTEN
F22A  D5      00337         PUSH DE               ;BEG.
F22B  CB7C    00338         BIT  7,H              ;! ASCII AUSGABE?
F22D  280B    00339         JRZ  MEM31-#          ;! JA
F22F  EB      00340         EX   DE,HL            ;SONST ADRESSE AUSGEBEN
F230  CD4AF1  00341         CALL OUTHL
F233  EB      00342         EX   DE,HL
F234  CD74F1  00343         CALL PRINT
F237          00344         DB   ' '
F239  00      00345         DB   0
F23A  0610    00346  MEM31: LD   B,MEMB           ;BYTEZAEHLER LADEN
F23C  1A      00347  MEM4:  LD   A,(DE)           ;AUSGABE HEX
F23D  13      00348         INC  DE
F23E  CD31F1  00349         CALL OUTH             ;BYTE AUSGEBEN
```

```
F241  2D      00350        DEC  L           ;CURSOR SETZEN?
F242  3E20    00351        LD   A,' '
F244  2001    00352        JRNZ MEM5-#      ;NEIN
F246  AF      00353        XOR  A           ;CODE FUER CURSOR
F247  CDC2FF  00354 MEM5:  CALL CO          ;SPACE ODER CURSOR AUSGEBEN
F24A  10F0    00355        DJNZ MEM4-#      ;WEITER
F24C  CB7C    00356        BIT  7,H         ;! ASCII AUSGABE?
F24E  2804    00357        JRZ  MEM51-#     ;! JA
              00358 ;LD A,' ' ;!
              00359 ;CALL CO ;!
F250  33      00360        INC  SP
F251  33      00361        INC  SP
F252  1812    00362        JR   MEM71-#
F254  0610    00363 MEM51: LD   B,MEMB      ;ZAEHLER
F256  D1      00364        POP  DE
F257  1A      00365 MEM6:  LD   A,(DE)      ;ASCII AUSGEBEN
F258  13      00366        INC  DE
F259  E67F    00367        AND  7FH         ;BIT 7 LOESCHEN
F25B  FE20    00368        CMP  ' '         ;TEST NICHT DARSTELLBAR?
F25D  3002    00369        JRNC MEM7-#      ;DOCH
F25F  3E2E    00370        LD   A,'.'       ;ANSONSTEN PUNKT AUSGEBEN
F261  CDC2FF  00371 MEM7:  CALL CO
F264  10F1    00372        DJNZ MEM6-#      ;WEITER
F266  CB7C    00373 MEM71: BIT  7,H         ;!
F268  2805    00374        JRZ  MEM72-#     ;!
F26A  3E1E    00375        LD   A,NL
F26C  CDC2FF  00376        CALL CO          ;NEUE ZEILE
F26F  C1      00377 MEM72: POP  BC
F270  10B7    00378        DJNZ MEM3-#      ;ZEILENZAHLER
              00379 ;EINGABE
```

```
F272  CBBC      00380          RES  7,H
F274  7C        00381          LD   A,H              ;MODE
F275  FE51      00382          CMP  'Q'
F277  2006      00383          JRNZ MEM73-#
F279  CDF8F0    00384          CALL INHEX            ;NEUE ADRESSE
F27C  C3CFF1    00385          JMP  MEM01
F27F  D5        00386  MEM73:  PUSH DE               ;ENDADR.+1 MERKEN
F280  5C        00387          LD   E,H              ;MODE FUER SPAETER
F281  1602      00388          LD   D,2              ;TETRADEN BYTEEINGABE
F283  CDCBFF    00389  MEM8:   CALL CI               ;ZEICHEN VON TASTATUR HOLEN
F286  FE1B      00390          CMP  ESC              ;UMSCHALTUNG HEX<==>ASCII EINGABE
F288  2015      00391          JRNZ MEM10-#
F28A  3E48      00392  MEM81:  LD   A,'H'
F28C  BB        00393          CMP  E                ;VERGLEICH MIT MODE
F28D  2002      00394          JRNZ MEM9-#           ;H ==>MODE
F28F  3E41      00395          LD   A,'A'
F291  5F        00396  MEM9:   LD   E,A              ;!
F292  3A10FF    00397          LD   A,(MEMMO)        ;!
F295  E680      00398          AND  80H              ;!
F297  B3        00399          OR   E                ;!
F298  3210FF    00400  MEM91:  LD   (MEMMO),A
F29B  D1        00401          POP  DE               ;KORR.
F29C  C3F0F1    00402          JMP  MEM1
F29F  01F0FF    00403  MEM10:  LD   BC,-MEMB         ;INC FUER CURSOR
F2A2  FE0B      00404          CMP  CURU             ;HOCH
F2A4  2836      00405          JRZ  MEM15-#
F2A6  0EFF      00406          LD   C,-1             ;BC=-1
F2A8  FE08      00407          CMP  CURL             ;LINKS
F2AA  2830      00408          JRZ  MEM15-#
F2AC  03        00409          INC  BC
```

```
F2AD  03       00410         INC  BC              ;BC=1
F2AE  FE09     00411         CMP  CURR            ;RECHTS
F2B0  282A     00412         JRZ  MEM15-#
F2B2  0E10     00413         LD   C,MEMB
F2B4  FE0A     00414         CMP  CURD            ;RUNTER
F2B6  2824     00415         JRZ  MEM15-#
F2B8  FE51     00416         CMP  'Q'             ;QUIT?
F2BA  28D5     00417         JRZ  MEM9-#
               00418 ;RAM BESCHREIBEN
F2BC  6F       00419 MEM11:  LD   L,A             ;ZEICHEN RETTEN
F2BD  7B       00420         LD   A,E             ;MODE
F2BE  FE41     00421         CMP  'A'             ;ASCII EINGABE?
F2C0  7D       00422         LD   A,L
F2C1  2810     00423         JRZ  MEM14-#         ;JA
               00424 ; HEX EINGABE
F2C3  CD20F1   00425         CALL CNVBN
F2C6  38BB     00426         JRC  MEM8-#          ;KEIN HEX,NEUE EINGABE
F2C8  210EFF   00427         LD   HL,ARG3         ;HILFSZELLEN
F2CB  ED6F     00428         RLD
F2CD  15       00429         DEC  D               ;TETRADENZAEHLER
F2CE  20B3     00430         JRNZ MEM8-#          ;NAECHSTES ZEICHEN
F2D0  3A0EFF   00431 MEM13:  LD   A,(ARG3)        ;NEUES BYTE
F2D3  2A0AFF   00432 MEM14:  LD   HL,(ARG1)       ;CURSOR
F2D6  77       00433         LD   (HL),A          ;=>RAM
F2D7  BE       00434         CMP  (HL)            ;ANGEKOMMEN?
F2D8  202B     00435         JRNZ MEM19-#         ;NEIN,RAM FEHLER
F2DA  0E01     00436         LD   C,1             ;BC=1;ADR INC
F2DC  D1       00437 MEM15:  POP  DE              ;ENDE+1
F2DD  2A0AFF   00438         LD   HL,(ARG1)       ;CURSOR
F2E0  09       00439         ADD  HL,BC           ;CURSOR+INC
```

```
F2E1  220AFF   00440         LD   (ARG1),HL       ;NEUER CURSOR
F2E4  CB78     00441         BIT  7,B             ;CURSOR RICHTUNG
F2E6  2805     00442         JRZ  MEM16-#         ;ADDITION
F2E8  EB       00443         EX   DE,HL           ;DE=CURSOR
F2E9  13       00444         INC  DE
F2EA  2A0CFF   00445         LD   HL,(ARG2)       ;ANFANG
F2ED  B7       00446 MEM16:  OR   A
F2EE  ED52     00447         SBC  HL,DE           ;ANF-CURS,CURS-ENDE
F2F0  2803     00448         JRZ  MEM17-#
F2F2  FAF0F1   00449         JPM  MEM1
F2F5  0D       00450 MEM17:  DEC  C               ;C=1?
F2F6  2002     00451         JRNZ MEM18-#
F2F8  0E0F     00452         LD   C,MEMB-1
F2FA  0C       00453 MEM18:  INC  C
F2FB  2A0CFF   00454         LD   HL,(ARG2)       ;NEUEN ANFANG BERECHNEN
F2FE  09       00455         ADD  HL,BC
F2FF  220CFF   00456         LD   (ARG2),HL
F302  C3F0F1   00457         JMP  MEM1
F305  CD74F1   00458 MEM19:  CALL PRINT
F308  0C       00459         DB   CLS
F309           00460         DB   'RAM ERROR
F313  00       00461         DB   0
F314  D1       00462         POP  DE
F315  C30AF2   00463         JMP  MEM2
               00464
               00465 ;*****************************
               00466 ;* MOVE                      *
               00467 ;*****************************
               00468
F318  EDFF     00469         DA   HEAD
```

```
F31A            00470           DB   'MOVE'
F31E  00        00471           NOP
F31F  CD00F1    00472           CALL ASHEX
F322  DA59F0    00473           JPC  ERROR
F325  220AFF    00474           LD   (ARG1),HL
F328  CD00F1    00475           CALL ASHEX
F32B  DA59F0    00476           JPC  ERROR
F32E  220CFF    00477           LD   (ARG2),HL
F331  CD00F1    00478           CALL ASHEX
F334  DA59F0    00479           JPC  ERROR
F337  44        00480           LD   B,H
F338  4D        00481           LD   C,L
F339  2A0AFF    00482           LD   HL,(ARG1)
F33C  ED5B0CFF  00483           LD   DE,(ARG2)
F340  E5        00484   MOVE:   PUSH HL
F341  AF        00485           XOR  A
F342  EB        00486           EX   DE,HL
F343  ED52      00487           SBC  HL,DE              ;ENDE-ANF=ANZAHL
F345  23        00488           INC  HL
F346  50        00489           LD   D,B
F347  59        00490           LD   E,C                ;DE=ZIEL
F348  44        00491           LD   B,H
F349  4D        00492           LD   C,L                ;BC=LAENGE
F34A  E1        00493           POP  HL                 ;HL=QUELLE
F34B  E5        00494           PUSH HL
F34C  B7        00495           OR   A                  ;CY=0
F34D  ED52      00496           SBC  HL,DE              ;MOVE-RICHTUNG
F34F  E1        00497           POP  HL
F350  3804      00498           JRC  MOVE1-#
F352  EDB0      00499           LDIR                    ;VORWAERTS UMLADEN
```

```
F354  1808     00500         JR   MOVE2-#
F356  0B       00501 MOVE1:  DEC  BC
F357  09       00502         ADD  HL,BC
F358  EB       00503         EX   DE,HL
F359  09       00504         ADD  HL,BC
F35A  EB       00505         EX   DE,HL
F35B  03       00506         INC  BC
F35C  EDB8     00507         LDDR
F35E  CDCBFF   00508 MOVE2:  CALL CI                 ;QUIT/MEM
F361  2A0AFF   00509         LD   HL,(ARG1)
F364  C3CFF1   00510         JMP  MEM01
               00511
               00512
               00513 ;******************************
               00514 ;* LOAD                       *
               00515 ;******************************
               00516
               00517 ;D7=1.LESEN,D6=UEBERLESEN
               00518 ;D5=BLOCK WAR FALSCH,D4=ALLES KORRIGIERT
               00519
F367  EDFF     00520         DA   HEAD
F369           00521         DB   'LOAD'
F36D  00       00522         NOP
F36E  CDB3F0   00523 LDEXT:  CALL GFINA
F371  CDD2F0   00524         CALL GFITY
F374  CD00F1   00525         CALL ASHEX
F377  3003     00526         JRNC LOAD-#
F379  210000   00527         LD   HL,0               ;KEIN OFFSET
F37C  E5       00528 LOAD:   PUSH HL
F37D  FDE1     00529         POP  IY
```

```
F37F  CD61FA   00530         CALL CINIT
F382  2A08FF   00531         LD   HL,(FITY)       ;OPTION
F385  1E00     00532         LD   E,0             ;E=FEHLER,D=KORREKTURZAHLER
F387  D9       00533         EXX
F388  1680     00534  LOD0:  LD   D,80H           ;1.LESEN
F38A  CB7A     00535  LOD1:  BIT  7,D
F38C  CA8CF4   00536         JPZ  LOD12           ;FILE NICHT LESBAR
F38F  D9       00537         EXX
F390  1600     00538         LD   D,0
F392  D9       00539         EXX
F393  CDADF4   00540  LOD2:  CALL LOD14           ;SYNCHRONISIEREN
F396  FE1E     00541         CMP  NL
F398  2003     00542         JRNZ LOD3-#
F39A  CDC8FF   00543         CALL CRI
F39D  FE01     00544  LOD3:  CMP  SOH
F39F  20F2     00545         JRNZ LOD2-#          ;KEIN SOH
F3A1  AF       00546         XOR  A
F3A2  4F       00547         LD   C,A             ;PRUEFSUMME=0
F3A3  5F       00548         LD   E,A             ;BLOCKZAHLER
F3A4  2100FF   00549         LD   HL,FINA
F3A7  0608     00550         LD   B,8
F3A9  CDC8FF   00551  LOD4:  CALL CRI             ;FILENAME VERGLEICHEN
F3AC  BE       00552         CMP  (HL)
F3AD  20DB     00553         JRNZ LOD1-#          ;NICHT GLEICH
F3AF  23       00554         INC  HL
F3B0  81       00555         ADD  C
F3B1  4F       00556         LD   C,A
F3B2  10F5     00557         DJNZ LOD4-#
F3B4  0618     00558         LD   B,24            ;KOPF LADEN
F3B6  CB7A     00559         BIT  7,D             ;1.LESEN
```

```
F3B8  2002      00560          JRNZ LOD5-#          ;JA
F3BA  CBF2      00561          SET  6,D             ;2.LESEN,KOPF UEBERLESEN
F3BC  CDBBF4    00562  LOD5:   CALL LOD16           ;BLOCK LESEN
F3BF  2804      00563          JRZ  LOD51-#         ;KOPF RICHTIG
F3C1  CB7A      00564          BIT  7,D
F3C3  20C3      00565          JRNZ LOD0-#
F3C5  CB7A      00566  LOD51:  BIT  7,D
F3C7  2829      00567          JRZ  LOD6-#          ;2.LESEN
F3C9  CD74F1    00568          CALL PRINT
F3CC  1E        00569          DB   NL
F3CD            00570          DB   'FILE FOUND
F3D8  00        00571          DB   0
F3D9  FDE5      00572          PUSH IY
F3DB  C1        00573          POP  BC              ;OFFSET
F3DC  2A0AFF    00574          LD   HL,(ARG1)
F3DF  09        00575          ADD  HL,BC
F3E0  220AFF    00576          LD   (ARG1),HL
F3E3  2A0CFF    00577          LD   HL,(ARG2)
F3E6  09        00578          ADD  HL,BC
F3E7  220CFF    00579          LD   (ARG2),HL
F3EA  78        00580          LD   A,B
F3EB  B1        00581          OR   C
F3EC  2804      00582          JRZ  LOD6-#          ;OFFSET=0
F3EE  AF        00583          XOR  A
F3EF  3208FF    00584          LD   (FITY),A        ;AUTOSTART UNTERDRUECKEN
                00585  ;DATENBLOECKE LESEN
F3F2  DD2110FF  00586  LOD6:   LD   IX,DATA
F3F6  2A0AFF    00587          LD   HL,(ARG1)       ;ANFANGSADR
F3F9  CDADF4    00588  LOD7:   CALL LOD14
F3FC  FE04      00589          CMP  EOT
```

```
F3FE  CA8CF4   00590         JPZ  LOD12
F401  FE03     00591         CMP  ETX                 ;ENDE?
F403  285E     00592         JRZ  LOD10-#             ;JA
F405  FE1D     00593         CMP  GS
F407  20F0     00594         JRNZ LOD7-#              ;KEIN RICHTIGER BLOCKANFANG
F409  0E00     00595         LD   C,0
F40B  CDC8FF   00596         CALL CRI                 ;BLOCKLAENGE
F40E  47       00597         LD   B,A
F40F  4F       00598         LD   C,A
F410  CBAA     00599         RES  5,D
F412  CB7A     00600         BIT  7,D
F414  2026     00601         JRNZ LOD8-#              ;1.LESEN
F416  CBF2     00602         SET  6,D                 ;FUER UBERLESEN,2.LESEN
F418  CB62     00603         BIT  4,D
F41A  2020     00604         JRNZ LOD8-#              ;ALLES KORR.
F41C  D5       00605         PUSH DE
F41D  7A       00606         LD   A,D
F41E  E60F     00607         AND  0FH
F420  DD5601   00608         LD   D,(IX+1)
               00609   ;WAR BLOCK SCHON FALSCH
F423  BA       00610         CMP  D
F424  D1       00611         POP  DE
F425  2015     00612         JRNZ LOD8-#              ;NEIN
F427  DD7E00   00613         LD   A,(IX)
F42A  BB       00614         CMP  E
F42B  200F     00615         JRNZ LOD8-#              ;NEIN
F42D  DD23     00616         INC  IX
F42F  DD23     00617         INC  IX
F431  D9       00618         EXX
F432  15       00619         DEC  D                   ;KORREKTURZAEHLER-1
```

```
F433  D9        00620         EXX
F434  2002      00621         JRNZ  LOD71-#          ;ALLES KORRIGIERT?
F436  CBE2      00622         SET   4,D              ;KEINE FALSCHEN BLOECKE MEHR
F438  CBB2      00623  LOD71: RES   6,D              ;FREIGABE RAM SCHREIBEN
F43A  CBEA      00624         SET   5,D              ;KORREKTURFLAG
F43C  CDBBF4    00625  LOD8:  CALL  LOD16            ;BLOCK LESEN
F43F  281F      00626         JRZ   LOD9-#           ;PRUEFSUMME OK
F441  CB6A      00627         BIT   5,D              ;WURDE BLOCK EBEN KORRIGIERT?
F443  2047      00628         JRNZ  LOD12-#          ;JA,FILE NICHT LESBAR
F445  CB7A      00629         BIT   7,D              ;NEIN
F447  2817      00630         JRZ   LOD9-#           ;FEHLER IGNORIEREN
F449  D9        00631         EXX
F44A  14        00632         INC   D
F44B  7A        00633         LD    A,D
F44C  D9        00634         EXX                    ;KORREKTURZAEHLER+1
F44D  FE08      00635         CMP   8
F44F  CA88F3    00636         JPZ   LOD0             ;ZU VIELE FEHLER,NOCHMAL VERSUCHEN
F452  DD7201    00637         LD    (IX+1),D         ;BLOCKNR. ABLEGEN
F455  DDCB01BE  00638         RES   7,(IX+1)         ;FLAG LOESCHEN
F459  DD7300    00639         LD    (IX+0),E
F45C  DD23      00640         INC   IX               ;ERRORPOINTER
F45E  DD23      00641         INC   IX
F460  13        00642  LOD9:  INC   DE               ;BLOCKZAEHLER+1
F461  1896      00643         JR    LOD7-#
                00644  ;ETX BEHANDLUNG
F463  7A        00645  LOD10: LD    A,D
F464  E680      00646         AND   80H              ;ZAHLER LOESCHEN,1/2-FLAG NICHT
F466  57        00647         LD    D,A
F467  2B        00648         DEC   HL               ;ENDADR.-1
F468  ED4B0CFF  00649         LD    BC,(ARG2)
```

```
F46C  ED42     00650         SBC  HL,BC           ;STIMMT ENDADR.?
F46E  C28AF3   00651         JPNZ LOD1            ;NEIN
F471  D9       00652         EXX
F472  7A       00653         LD   A,D
F473  B7       00654         OR   A               ;TRATEN FEHLER AUF?
F474  2805     00655         JRZ  LOD11-#         ;NEIN
F476  D9       00656         EXX
F477  55       00657         LD   D,L             ;2.LESEN
F478  C393F3   00658         JMP  LOD2
F47B  CD9BF4   00659  LOD11: CALL LOD13           ;FEHLERAUSGABE
F47E  3A08FF   00660         LD   A,(FITY)
F481  FE50     00661         CMP  'P'
F483  C0       00662         RNZ
F484  7D       00663         LD   A,L
F485  FE51     00664         CMP  'Q'
F487  C8       00665         RZ
F488  2A0EFF   00666         LD   HL,(ARG3)
F48B  E9       00667         JMP  (HL)            ;AUTOSTART
F48C  D9       00668  LOD12: EXX
F48D  CD74F1   00669         CALL PRINT
F490           00670         DB   ' BAD FILE
F49A  00       00671         DB   0
F49B  7B       00672  LOD13: LD   A,E
F49C  B7       00673         OR   A
F49D  C8       00674         RZ
F49E  CD31F1   00675         CALL OUTH
F4A1  CD74F1   00676         CALL PRINT
F4A4           00677         DB   ' ERRORS'
F4AB  00       00678         DB   0
F4AC  C9       00679         RET
```

```
                    00680  ;SYNCHRONISIEREN
F4AD  CDC8FF        00681  LOD14:  CALL CRI
F4B0  3C            00682          INC  A
F4B1  20FA          00683          JRNZ LOD14-#
F4B3  CDC8FF        00684  LOD15:  CALL CRI
F4B6  FEFF          00685          CMP  OFFH
F4B8  28F9          00686          JRZ  LOD15-#
F4BA  C9            00687          RET
                    00688  ;BLOCK LESEN
F4BB  CDC8FF        00689  LOD16:  CALL CRI
F4BE  CB72          00690          BIT  6,D
F4C0  2001          00691          JRNZ LOD17-#             ;UEBERLESEN
F4C2  77            00692          LD   (HL),A
F4C3  81            00693  LOD17:  ADD  C
F4C4  4F            00694          LD   C,A
F4C5  23            00695          INC  HL
F4C6  10F3          00696          DJNZ LOD16-#
F4C8  CDC8FF        00697          CALL CRI                 ;PRUEFSUMME
F4CB  81            00698          ADD  C
F4CC  CBB2          00699          RES  6,D
F4CE  C8            00700          RZ                       ;PRUEFSUMME OK
F4CF  D9            00701          EXX
F4D0  1C            00702          INC  E
F4D1  D9            00703          EXX                      ;FEHLERZAHLER+1
F4D2  C9            00704          RET
                    00705
                    00706  ;****************************
                    00707  ;* SAVE                     *
                    00708  ;****************************
                    00709
```

```
F4D3  EDFF     00710          DA   HEAD
F4D5           00711          DB   'SAVE'
F4D9  00       00712          NOP
F4DA  CDB3F0   00713  SAEXT:  CALL GFINA                ;FILENAME HOLEN
F4DD  CDD2F0   00714          CALL GFITY                ;FILETYP
F4E0  CD00F1   00715          CALL ASHEX                ;STARTADRESSE
F4E3  DA59F0   00716          JPC  ERROR
F4E6  220AFF   00717          LD   (ARG1),HL
F4E9  CD00F1   00718          CALL ASHEX                ;ENDADRESSE
F4EC  DA59F0   00719          JPC  ERROR
F4EF  220CFF   00720          LD   (ARG2),HL
F4F2  CD00F1   00721          CALL ASHEX                ;EINTRITTSPUNKT
F4F5  3003     00722          JRNC SAVE-#
F4F7  211EF0   00723          LD   HL,RSTAR             ;FEHLT,DESHALB RSTAR
F4FA  220EFF   00724  SAVE:   LD   (ARG3),HL
F4FD  CDE1F0   00725          CALL GCOM
F500  CD61FA   00726          CALL CINIT
F503  2189F5   00727          LD   HL,SAVEE
F506  E5       00728          PUSH HL
F507  211AF5   00729          LD   HL,SAVE0
F50A  E5       00730          PUSH HL                   ;RETURN=>SAVE0
F50B  E5       00731          PUSH HL
F50C  0E1E     00732          LD   C,NL                 ;FILEANFANG
F50E  3A08FF   00733          LD   A,(FITY)
F511  FE20     00734          CMP  ' '
F513  2005     00735          JRNZ SAVE0-#
F515  3E50     00736          LD   A,'P'                ;KEINE ANGABE
F517  3208FF   00737          LD   (FITY),A
F51A  0614     00738  SAVE0:  LD   B,20
F51C  3EFF     00739  SAVE1:  LD   A,0FFH               ;SYNCHRONZEICHEN
```

```
F51E  CDBEFF    00740          CALL CPO
F521  10F9      00741          DJNZ SAVE1-#
F523  79        00742          LD   A,C
F524  CDBEFF    00743          CALL CPO             ;NL ODER SYNCHRONZEICHEN
F527  3E01      00744          LD   A,SOH           ;START KOPF
F529  CDBEFF    00745          CALL CPO
                00746   ;KOPF AUSGEBEN
                00747   ;SOH/FILENAME/TT/AA/EE/SS/................/PS/STX
F52C  2100FF    00748          LD   HL,FINA
F52F  0620      00749          LD   B,32
F531  0E00      00750          LD   C,0
F533  7E        00751   SAVE2: LD   A,(HL)
F534  CDBEFF    00752          CALL CPO
F537  7E        00753          LD   A,(HL)
F538  81        00754          ADD  C               ;PRUEFSUMME
F539  4F        00755          LD   C,A
F53A  23        00756          INC  HL
F53B  10F6      00757          DJNZ SAVE2-#
F53D  ED44      00758          NEG
F53F  CDBEFF    00759          CALL CPO             ;PUEFSUMME
F542  3E02      00760          LD   A,STX           ;KOPF-ENDE
F544  CDBEFF    00761          CALL CPO
F547  ED5B0AFF  00762          LD   DE,(ARG1)       ;ANF.ADR.
F54B  0606      00763   SAVE3: LD   B,6
F54D  3EFF      00764   SAVE4: LD   A,0FFH          ;SYNC.
F54F  CDBEFF    00765          CALL CPO
F552  10F9      00766          DJNZ SAVE4-#
F554  2A0CFF    00767          LD   HL,(ARG2)       ;ENDADR.
F557  23        00768          INC  HL
F558  ED52      00769          SBC  HL,DE           ;BLOCKLAENGE BERECHNEN
```

```
F55A  2826     00770         JRZ  SAVE8-#          ;FERTIG
F55C  AF       00771         XOR  A
F55D  B4       00772         OR   H
F55E  2004     00773         JRNZ SAVE5-#
F560  85       00774         ADD  L
F561  F266F5   00775         JPP  SAVE6
F564  2E80     00776  SAVE5: LD   L,80H
F566  45       00777  SAVE6: LD   B,L              ;BYTEZAEHLER
F567  3E1D     00778         LD   A,GS             ;START BLOCK
F569  CDBEFF   00779         CALL CPO
F56C  78       00780         LD   A,B
F56D  4F       00781         LD   C,A
F56E  CDBEFF   00782         CALL CPO              ;BLOCKLAENGE
F571  1A       00783  SAVE7: LD   A,(DE)
F572  CDBEFF   00784         CALL CPO
F575  1A       00785         LD   A,(DE)
F576  13       00786         INC  DE
F577  81       00787         ADD  C
F578  4F       00788         LD   C,A
F579  10F6     00789         DJNZ SAVE7-#
F57B  ED44     00790         NEG
F57D  CDBEFF   00791         CALL CPO              ;PRUEFSUMME
F580  18C9     00792         JR   SAVE3-#
F582  3E03     00793  SAVE8: LD   A,ETX
F584  0EFF     00794         LD   C,0FFH           ;2.AUFZEICHNUNG
F586  C3BEFF   00795         JMP  CPO
F589  0620     00796  SAVEE: LD   B,32
F58B  3EFF     00797  SAVEF: LD   A,0FFH
F58D  CDBEFF   00798         CALL CPO
F590  10F9     00799         DJNZ SAVEF-#
```

```
F592  3E04     00800        LD   A,EOT
F594  C3BEFF   00801        JMP  CPO
               00802
               00803
               00804  ;*****************************
               00805  ;*  IN OUT                   *
               00806  ;*****************************
               00807
F597  EDFF     00808        DA   HEAD
F599           00809        DB   'IN'
F59B  00       00810        NOP
F59C  CD00F1   00811        CALL ASHEX
F59F  4D       00812        LD   C,L
F5A0  ED78     00813        IN   A
F5A2  C331F1   00814        JMP  OUTH
               00815
F5A5  EDFF     00816        DA   HEAD
F5A7           00817        DB   'OUT'
F5AA  00       00818        NOP
F5AB  CD00F1   00819        CALL ASHEX              ;PORTADRESSE HOLEN
F5AE  4D       00820        LD   C,L
F5AF  CD00F1   00821        CALL ASHEX              ;WERT HOLEN
F5B2  ED69     00822        OUT  L
F5B4  C9       00823        RET
               00824
               00825  ;*****************************
               00826  ;* FILL                      *
               00827  ;*****************************
               00828
F5B5  EDFF     00829        DA   HEAD
```

```
F5B7              00830          DB   'FILL'
F5BB   00         00831          NOP
F5BC   CD00F1     00832          CALL ASHEX
F5BF   DA59F0     00833          JPC  ERROR
F5C2   220AFF     00834          LD   (ARG1),HL
F5C5   CD00F1     00835          CALL ASHEX
F5C8   DA59F0     00836          JPC  ERROR
F5CB   220CFF     00837          LD   (ARG2),HL
F5CE   CD00F1     00838          CALL ASHEX
F5D1   4D         00839          LD   C,L               ;FUELLBYTE
F5D2   2A0AFF     00840          LD   HL,(ARG1)
F5D5   ED5B0CFF   00841          LD   DE,(ARG2)
F5D9   3002       00842          JRNC FILL1-#
F5DB   0E00       00843          LD   C,0               ;FUELLBYTE 00 FESTLEGEN
F5DD   71         00844  FILL1:  LD   (HL),C
F5DE   E5         00845          PUSH HL
F5DF   AF         00846          XOR  A
F5E0   EB         00847          EX   DE,HL
F5E1   ED52       00848          SBC  HL,DE
F5E3   44         00849          LD   B,H
F5E4   4D         00850          LD   C,L
F5E5   E1         00851          POP  HL
F5E6   54         00852          LD   D,H
F5E7   5D         00853          LD   E,L
F5E8   13         00854          INC  DE
F5E9   EDB0       00855          LDIR
F5EB   C9         00856          RET
                  00857
                  00858
                  00859  ;******************************
```

```
                   00860  ;* MEMSIZE                      *
                   00861  ;********************************
                   00862
F5EC  EDFF         00863        DA   HEAD
F5EE               00864        DB   'SIZE'
F5F2  00           00865        NOP
F5F3  CD74F1       00866        CALL PRINT
F5F6               00867        DB   'SIZE: '
F5FB  00           00868        DB   0
F5FC  2AB1FF       00869        LD   HL,(MSIZE)
F5FF  CD4AF1       00870        CALL OUTHL
F602  CD74F1       00871        CALL PRINT
F605               00872        DB   ' RAMTOP:'
F60D  00           00873        DB   0
F60E  CD32FA       00874        CALL MEMCH              ;RAMTEST
F611  CD4AF1       00875        CALL OUTHL
F614  22B1FF       00876        LD   (MSIZE),HL
F617  CDF8F0       00877        CALL INHEX
F61A  D8           00878        RC
F61B  22B1FF       00879        LD   (MSIZE),HL
F61E  C9           00880        RET
                   00881
                   00882
                   00883  ;********************************
                   00884  ;* TESTPROGRAMME                 *
                   00885  ;********************************
                   00886
                   00887  ;NMI VEKTOR EINTRAGEN
F61F  EDFF         00888        DA   HEAD
F621               00889        DB   'NMI'
```

```
F624  00        00890          NOP
F625  216600    00891  NMIV:   LD   HL,66H
F628  36C3      00892          LD   (HL),0C3H          ;JMP
F62A  23        00893          INC  HL
F62B  3631      00894          LD   (HL),L(BREAK)
F62D  23        00895          INC  HL
F62E  36F6      00896          LD   (HL),H(BREAK)
F630  C9        00897          RET
                00898
F631  CD55F6    00899  BREAK:  CALL UPUSH              ;BEI NMI
F634  E1        00900          POP  HL                 ;PC
F635  ED73ABFF  00901          LD   (USP),SP
F639  22ADFF    00902          LD   (UPC),HL
F63C  31FEFE    00903          LD   SP,STACK
F63F  CD74F1    00904          CALL PRINT
F642            00905          DB   'BREAK AT '
F64B  00        00906          DB   0
F64C  CD4AF1    00907          CALL OUTHL
F64F  216CF0    00908          LD   HL,MAIN
F652  E5        00909          PUSH HL
F653  ED45      00910          RETN
                00911
                00912  ;CPU AUSRAEUMEN
F655  ED73AFFF  00913  UPUSH:  LD   (SSP),SP
F659  31A9FF    00914          LD   SP,HIF
F65C  FDE5      00915          PUSH IY
F65E  DDE5      00916          PUSH IX
F660  D9        00917          EXX                     ;ERST ZWEITREGISTER
F661  08        00918          EXAF
F662  E5        00919          PUSH HL
```

```
F663  D5        00920          PUSH DE
F664  C5        00921          PUSH BC
F665  F5        00922          PUSH AF
F666  D9        00923          EXX                    ;DANN HAUPTREGISTER
F667  08        00924          EXAF
F668  E5        00925          PUSH HL
F669  D5        00926          PUSH DE
F66A  C5        00927          PUSH BC
F66B  F5        00928          PUSH AF
F66C  ED57      00929          LD   A,I
F66E  32A9FF    00930          LD   (HIF),A
F671  181C      00931          JR   UPOP1-#
                00932
                00933 ;CPU EINRAEUMEN
F673  ED73AFFF  00934 UPOP:    LD   (SSP),SP
F677  3195FF    00935          LD   SP,UREG
F67A  3AA9FF    00936          LD   A,(HIF)
F67D  ED47      00937          LD   I,A
F67F  F1        00938          POP  AF
F680  C1        00939          POP  BC
F681  D1        00940          POP  DE
F682  E1        00941          POP  HL
F683  D9        00942          EXX
F684  08        00943          EXAF
F685  F1        00944          POP  AF
F686  C1        00945          POP  BC
F687  D1        00946          POP  DE
F688  E1        00947          POP  HL
F689  DDE1      00948          POP  IX
F68B  FDE1      00949          POP  IY
```

```
F68D  D9        00950          EXX
F68E  08        00951          EXAF
F68F  ED7BAFFF  00952  UPOP1:  LD   SP,(SSP)
F693  C9        00953          RET
                00954
F694  EDFF      00955          DA   HEAD
F696            00956          DB   'GO'
F698  00        00957          NOP
F699  1B        00958          DEC  DE
F69A  CD00F1    00959          CALL ASHEX               ;ADR.
F69D  383A      00960          JRC  G01-#               ;KEINE ADR
                00961
F69F  E5        00962          PUSH HL                  ;ADR. RETTEN
F6A0  CD00F1    00963          CALL ASHEX               ;BREAKP.?
F6A3  3829      00964          JRC  G02-#               ;KEIN BREAKPOINT
                00965
F6A5  3ACFFF    00966          LD   A,(BRKPT+1)
F6A8  FEF0      00967          CMP  H(START)            ;IST BRKPT SCHON GESETZT?
F6AA  280E      00968          JRZ  G001-#              ;NEIN
F6AC  E5        00969          PUSH HL                  ;BREAKPOINT RETTEN
F6AD  2ACEFF    00970          LD   HL,(BRKPT)          ;ALTER BRKPT
F6B0  7E        00971          LD   A,(HL)              ;STEHT DA NOCH RST 38?
F6B1  FEFF      00972          CMP  0FFH
F6B3  2004      00973          JRNZ G00-#               ;NEIN/ALSO KEIN ALTER BREAKPOINT
F6B5  3AD0FF    00974          LD   A,(BRKPT+2)         ;AUSTAUSCHEN
F6B8  77        00975          LD   (HL),A
F6B9  E1        00976  G00:    POP  HL                  ;NEUER BRKPT ZURUECK
F6BA  22CEFF    00977  G001:   LD   (BRKPT),HL          ;NEUEN BRKPT SPEICHERN
F6BD  7E        00978          LD   A,(HL)              ;BYTE RETTEN
F6BE  32D0FF    00979          LD   (BRKPT+2),A
```

```
F6C1  36FF      00980         LD    (HL),0FFH        ;RST38 EINTRAGEN
F6C3  213800    00981         LD    HL,38H
F6C6  36C3      00982         LD    (HL),0C3H        ;JMP
F6C8  23        00983         INC   HL
F6C9  36FC      00984         LD    (HL),L(BRK)
F6CB  23        00985         INC   HL
F6CC  36F6      00986         LD    (HL),H(BRK)
F6CE  E1        00987  G02:   POP   HL
F6CF  31C0FE    00988         LD    SP,USTCK         ;USERSTACK
F6D2  116CF0    00989         LD    DE,MAIN          ;RUECKSPRUNG
F6D5  D5        00990         PUSH  DE
F6D6  E5        00991         PUSH  HL
F6D7  189A      00992         JR    UPOP-#           ;SPRUNG ADR.
F6D9  FE50      00993  G01:   CMP   'P'
F6DB  2011      00994         JRNZ  G03-#
F6DD  13        00995         INC   DE
F6DE  1A        00996         LD    A,(DE)
F6DF  FE43      00997         CMP   'C'
F6E1  C259F0    00998         JPNZ  ERROR
F6E4  2AADFF    00999         LD    HL,(UPC)
F6E7  ED7BABFF  01000         LD    SP,(USP)
F6EB  E5        01001         PUSH  HL
F6EC  1885      01002         JR    UPOP-#
F6EE  2ACEFF    01003  G03:   LD    HL,(BRKPT)
F6F1  7E        01004         LD    A,(HL)
F6F2  FEFF      01005         CMP   0FFH
F6F4  C259F0    01006         JPNZ  ERROR
F6F7  3AD0FF    01007         LD    A,(BRKPT+2)
F6FA  77        01008         LD    (HL),A
F6FB  C9        01009         RET
```

```
                 01010
                 01011
F6FC  CD55F6     01012  BRK:    CALL UPUSH
F6FF  E1         01013          POP  HL
F700  2B         01014          DEC  HL                  ;ALTER PC
F701  ED73ABFF   01015          LD   (USP),SP
F705  22ADFF     01016          LD   (UPC),HL
F708  31FEFE     01017          LD   SP,STACK
F70B  3AD0FF     01018          LD   A,(BRKPT+2)         ;BYTE WIEDER EINTRAGEN
F70E  77         01019          LD   (HL),A
F70F  2100F0     01020          LD   HL,START            ;BREAKPOINT LOESCHEN
F712  22CEFF     01021          LD   (BRKPT),HL
F715  216CF0     01022          LD   HL,MAIN
F718  E5         01023          PUSH HL
                 01024
F719  CD74F1     01025  REGDP:  CALL PRINT
F71C             01026          DB   ' AF   BC   DE   HL   AF'
F733  27         01027          DB   27H
F734             01028          DB   '  BC'
F738  27         01029          DB   27H
F739             01030          DB   '  DE'
F73D  27         01031          DB   27H
F73E             01032          DB   '  HL'
F742  27         01033          DB   27H
F743             01034          DB   '  IX   IY  I    SP    PC'
F75B  1E         01035          DB   NL
F75C  00         01036          DB   0
F75D  060A       01037          LD   B,10
F75F  DD2195FF   01038          LD   IX,UREG
F763  DD6E00     01039  REGD1:  LD   L,(IX)
```

```
F766  DD6601   01040          LD   H,(IX+1)
F769  CD4AF1   01041          CALL OUTHL
F76C  3E20     01042          LD   A,' '
F76E  CDC2FF   01043          CALL CO
F771  DD23     01044          INC  IX
F773  DD23     01045          INC  IX
F775  10EC     01046          DJNZ REGD1-#
F777  3AA9FF   01047          LD   A,(HIF)
F77A  CD31F1   01048          CALL OUTH
F77D  CD74F1   01049          CALL PRINT
F780           01050          DB   '  '
F782  00       01051          DB   0
F783  2AABFF   01052          LD   HL,(USP)
F786  CD4AF1   01053          CALL OUTHL
F789  CD74F1   01054          CALL PRINT
F78C           01055          DB   '  '
F78E  00       01056          DB   0
F78F  2AADFF   01057          LD   HL,(UPC)
F792  CD4AF1   01058          CALL OUTHL
F795  C9       01059          RET
F796  C319F7   01060  REGD:   JMP  REGDP
               01061
F799  EDFF     01062          DA   HEAD
F79B           01063          DB   'REG'
F79E  00       01064          NOP
F79F  CD55F1   01065  REG1:   CALL FNEXT
F7A2  38F2     01066          JRC  REGD-#                ;WEIL CR
F7A4  2109F8   01067          LD   HL,RTAB
F7A7  010800   01068          LD   BC,8
F7AA  EDB1     01069          CPIR                       ;AFBCDEHL SUCHEN
```

```
F7AC  201A     01070        JRNZ REG4-#          ;DOPPELREG.
F7AE  018B07   01071        LD   BC,UREG-RTAB-1
F7B1  09       01072        ADD  HL,BC           ;ADRESSE DES REG.
F7B2  13       01073        INC  DE
F7B3  1A       01074        LD   A,(DE)
F7B4  FE27     01075        CMP  27H             ;STRICHREGISTER
F7B6  2005     01076        JRNZ REG2-#          ;NEIN
F7B8  13       01077        INC  DE
F7B9  3E08     01078        LD   A,8             ;WEIL STRICHREGISTER
F7BB  85       01079        ADD  L               ;ADR KORRIGIEREN
F7BC  6F       01080        LD   L,A
F7BD  E5       01081 REG2:  PUSH HL
F7BE  CD00F1   01082 REG3:  CALL ASHEX           ;BYTE HOLEN
F7C1  7D       01083        LD   A,L
F7C2  E1       01084        POP  HL
F7C3  38D1     01085        JRC  REGD-#          ;KEIN HEX
F7C5  77       01086        LD   (HL),A          ;EINTRAGEN
F7C6  18D7     01087        JR   REG1-#
F7C8  FE49     01088 REG4:  CMP  'I'             ;DOPPELREGISTER
F7CA  2014     01089        JRNZ REG5-#
F7CC  13       01090        INC  DE
F7CD  1A       01091        LD   A,(DE)
F7CE  FE58     01092        CMP  'X'
F7D0  21A5FF   01093        LD   HL,HIX          ;IX REGISTER
F7D3  2825     01094        JRZ  REG7-#
F7D5  FE59     01095        CMP  'Y'
F7D7  23       01096        INC  HL
F7D8  23       01097        INC  HL
F7D9  281F     01098        JRZ  REG7-#          ;IY REGITER
F7DB  21A9FF   01099        LD   HL,HIF          ;INTERRUPTREGISTER
```

```
F7DE  18DD      01100          JR    REG2-#
F7E0  FE50      01101  REG5:   CMP   'P'
F7E2  200B      01102          JRNZ  REG6-#
F7E4  13        01103          INC   DE
F7E5  1A        01104          LD    A,(DE)
F7E6  FE43      01105          CMP   'C'
F7E8  20AC      01106          JRNZ  REGD-#            ;FALSCHE EINGABE
F7EA  21ADFF    01107          LD    HL,UPC            ;PC
F7ED  180B      01108          JR    REG7-#
F7EF  FE53      01109  REG6:   CMP   'S'
F7F1  13        01110          INC   DE
F7F2  1A        01111          LD    A,(DE)
F7F3  FE50      01112          CMP   'P'
F7F5  209F      01113          JRNZ  REGD-#            ;FALSCHE EINGABE
F7F7  21ABFF    01114          LD    HL,USP            ;STACKPOINTER
F7FA  E5        01115  REG7:   PUSH  HL
F7FB  13        01116          INC   DE
F7FC  CD00F1    01117          CALL  ASHEX             ;WORT HOLEN
F7FF  44        01118          LD    B,H
F800  4D        01119          LD    C,L
F801  E1        01120          POP   HL
F802  3892      01121          JRC   REGD-#            ;FALLS KEINE HEX-EINGABE
F804  71        01122          LD    (HL),C            ;EINTRAGEN
F805  23        01123          INC   HL
F806  70        01124          LD    (HL),B
F807  1896      01125          JR    REG1-#
                01126
F809            01127  RTAB:   DB    'FACBEDLH'
                01128
                01129
```

```
                  01130 ;******************************
                  01131 ;* HELP                        *
                  01132 ;******************************
                  01133
F811  EDFF        01134         DA   HEAD
F813              01135         DB   'HELP'
F817  00          01136         NOP
F818  210000      01137         LD   HL,0
F81B  010000      01138         LD   BC,0
F81E  3E1E        01139         LD   A,NL
F820  CDC2FF      01140         CALL CO
F823  CD74F1      01141 HELP1:  CALL PRINT
F826              01142         DB   '   '
F829  00          01143         DB   0
F82A  3EED        01144 HELP2:  LD   A,L(HEAD)
F82C  EDB1        01145         CPIR                ;SUCHE NACH ED
F82E  C0          01146         RNZ
F82F  3EFF        01147         LD   A,H(HEAD)
F831  BE          01148         CMP  (HL)           ;FOLGT FF?
F832  20F6        01149         JRNZ HELP2-#        ;NEIN
F834  E5          01150         PUSH HL
F835  C5          01151         PUSH BC
F836  0610        01152         LD   B,16           ;TEST,OB NACH SPAETESTENS
F838  7E          01153 HELP3:  LD   A,(HL)         ;16 ZEICHEN EIN NOP FOLGT
F839  23          01154         INC  HL
F83A  B7          01155         OR   A
F83B  2806        01156         JRZ  HELP4-#        ;JA,AUSGABE
F83D  10F9        01157         DJNZ HELP3-#
F83F  C1          01158         POP  BC
F840  E1          01159         POP  HL
```

```
F841  18E7     01160          JR    HELP2-#
F843  C1       01161  HELP4:  POP   BC                            ;AUSGABE
F844  E1       01162          POP   HL
F845  23       01163  HELP5:  INC   HL
F846  0B       01164          DEC   BC
F847  7E       01165          LD    A,(HL)
F848  B7       01166          OR    A                             ;TEST ENDE
F849  28D8     01167          JRZ   HELP1-#
F84B  CDC2FF   01168          CALL  CO
F84E  18F5     01169          JR    HELP5-#
               01170
               01171
               01172
               01173  ;******************************
               01174  ;* READMC                     *
               01175  ;******************************
               01176
F850  EDFF     01177          DA    HEAD
F852           01178          DB    'READ'
F856  00       01179          NOP
F857  CD7BFA   01180          CALL  RINIT
F85A  CDD2F0   01181          CALL  GFITY
F85D  CD00F1   01182          CALL  ASHEX
F860  110000   01183          LD    DE,0
F863  3801     01184          JRC   READ1-#
F865  EB       01185          EX    DE,HL
F866  CDC5FF   01186  READ1:  CALL  RI                            ;DOPPELPUNKT SUCHEN
F869  FE3A     01187          CMP   3AH
F86B  20F9     01188          JRNZ  READ1-#
F86D  0E00     01189          LD    C,0                           ;PRUEFSUMME =0 SETZEN
```

```
F86F  CDCEF8   01190          CALL READ3         ;BLOCKLAENGE
               01191
F872  B7       01192          OR   A             ;ENDE?
F873  C8       01193          RZ                 ;FERTIG
F874  47       01194          LD   B,A
F875  CDCEF8   01195          CALL READ3         ;HI DER ADR.
F878  67       01196          LD   H,A
F879  CDCEF8   01197          CALL READ3         ;LO
F87C  6F       01198          LD   L,A
F87D  3A08FF   01199          LD   A,(FITY)
F880  FE41     01200          CMP  'A'
F882  2017     01201          JRNZ REA12-#
F884  C5       01202          PUSH BC
F885  D5       01203          PUSH DE
F886  E5       01204          PUSH HL
F887  CD4AF1   01205          CALL OUTHL
F88A  3E20     01206          LD   A,' '
F88C  3208FF   01207          LD   (FITY),A
F88F  CDF8F0   01208          CALL INHEX
F892  D1       01209          POP  DE
F893  3803     01210          JRC  REA11-#
F895  ED52     01211          SBC  HL,DE
F897  E3       01212          EX   (SP),HL
F898  E1       01213 REA11:   POP  HL
F899  EB       01214          EX   DE,HL
F89A  C1       01215          POP  BC
F89B  19       01216 REA12:   ADD  HL,DE         ;ADR+OFFSET
F89C  CDCEF8   01217          CALL READ3
F89F  CDCEF8   01218 READ2:   CALL READ3         ;DATEN
F8A2  77       01219          LD   (HL),A
```

```
F8A3  23        01220          INC  HL
F8A4  10F9      01221          DJNZ READ2-#
F8A6  CDCBF8    01222          CALL READ3              ;PRUEFSUMMPE
F8A9  28BB      01223          JRZ  READ1-#            ;OK
F8AB  CD74F1    01224          CALL PRINT
F8AE            01225          DB   'CHECKSUM ERROR
F8BD  00        01226          DB   0
F8BE  CD4AF1    01227          CALL OUTHL
F8C1  CDCBFF    01228          CALL CI                 ;HALT BIS TASTE GEDRUECKT
F8C4  FE51      01229          CMP  'Q'                ;QUIT?
F8C6  C8        01230          RZ                      ;JA
F8C7  3E1B      01231          LD   A,NL
F8C9  CDC2FF    01232          CALL CO
F8CC  1898      01233          JR   READ1-#
F8CE  D5        01234  READ3:  PUSH DE                 ;BYTE VOM LESER
F8CF  CDC5FF    01235          CALL RI
F8D2  CD20F1    01236          CALL CNVBN
F8D5  07        01237          RLCA
F8D6  07        01238          RLCA
F8D7  07        01239          RLCA
F8D8  07        01240          RLCA
F8D9  57        01241          LD   D,A
F8DA  CDC5FF    01242          CALL RI
F8DD  CD20F1    01243          CALL CNVBN
F8E0  B2        01244          OR   D                  ;BYTE ZUSAMMENFUEGEN
F8E1  57        01245          LD   D,A                ;RETTEN
F8E2  81        01246          ADD  C                  ;PRUEFSUMME
F8E3  4F        01247          LD   C,A                ;NACH C ZURUECK
F8E4  7A        01248          LD   A,D
F8E5  D1        01249          POP  DE
```

```
F8E6  C9        01250          RET
                01251
                01252  ;*****************************
                01253  ;*   CHECKSUM               *
                01254  ;*****************************
                01255
F8E7  EDFF      01256          DA    HEAD
F8E9            01257          DB    'CHECKSUM'
F8F1  00        01258          NOP
F8F2  CD00F1    01259          CALL  ASHEX
F8F5  DA59F0    01260          JPC   ERROR
F8F8  E5        01261          PUSH  HL
F8F9  CD00F1    01262          CALL  ASHEX
F8FC  EB        01263          EX    DE,HL
F8FD  13        01264          INC   DE            ;ENDADRESSE+1
F8FE  E1        01265          POP   HL            ;STARTADR.
F8FF  DA59F0    01266          JPC   ERROR
F902  DD210000  01267  CHECK:  LD    IX,0          ;IX DIENT ALS AKKU
F906  0600      01268          LD    B,0
F908  4E        01269  CHCK1:  LD    C,(HL)
F909  DD09      01270          ADD   IX,BC         ;SUMMIEREN
F90B  23        01271          INC   HL
F90C  E5        01272          PUSH  HL
F90D  AF        01273          XOR   A             ;CY=0
F90E  ED52      01274          SBC   HL,DE         ;ENDADRESSE ERRREICHT?
F910  E1        01275          POP   HL
F911  38F5      01276          JRC   CHCK1-#       ;NEIN,WEITER
F913  DDE5      01277          PUSH  IX            ;ERGEBNIS
F915  E1        01278          POP   HL            ;NACH HL FUER AUSGABE
F916  C34AF1    01279          JMP   OUTHL         ;AUSGABE UND RETURN
```

```
                     01280
                     01281  ;******************************
                     01282  ;*  CALCULATE                 *
                     01283  ;******************************
                     01284
F919   EDFF          01285          DA   HEAD
F91B                 01286          DB   '='
F91C   00            01287          NOP
F91D   CD00F1        01288          CALL ASHEX
F920   E5            01289          PUSH HL
F921   E5            01290          PUSH HL
F922   E5            01291          PUSH HL
F923   CD00F1        01292          CALL ASHEX
F926   EB            01293          EX   DE,HL
F927   E1            01294          POP  HL                ;1.WERT
F928   19            01295          ADD  HL,DE             ;SUMME AUSRECHNEN
F929   CD74F1        01296          CALL PRINT             ;UND AUSGEBEN
F92C                 01297          DB   'SUM:'
F930   00            01298          DB   0
F931   CD4AF1        01299          CALL OUTHL
F934   E1            01300          POP  HL                ;DIFFERENZ AUSRECHNEN
F935   B7            01301          OR   A                 ;CY=0
F936   ED52          01302          SBC  HL,DE
F938   CD74F1        01303          CALL PRINT             ;UND AUSGEBEN
F93B                 01304          DB   '  DIF:'
F941   00            01305          DB   0
F942   CD4AF1        01306          CALL OUTHL
F945   E1            01307          POP  HL
F946   EB            01308          EX   DE,HL
F947   B7            01309          OR   A
```

```
F948  ED52      01310         SBC  HL,DE
F94A  2B        01311         DEC  HL
F94B  2B        01312         DEC  HL
F94C  CD74F1    01313         CALL PRINT
F94F            01314         DB   '  REL:
F955  00        01315         DB   0
F956  29        01316         ADD  HL,HL           ;TEST BEREICHS UEBERSCHREITUNG
F957  7C        01317         LD   A,H             ;BEI RELATIVEN SPRUENGEN
F958  CE00      01318         ADC  0
F95A  2005      01319         JRNZ CALC1-#         ;ZU WEIT
F95C  7D        01320         LD   A,L
F95D  1F        01321         RRA                  ;/2
F95E  C331F1    01322         JMP  OUTH            ;AKKU AUSGEBEN UND RETURN
F961  CD74F1    01323 CALC1:  CALL PRINT           ;WENN ZU WEIT
F964            01324         DB   '**'
F966  00        01325         DB   0
F967  C9        01326         RET
                01327
                01328 ;*****************************
                01329 ;*      FIND                *
                01330 ;*****************************
                01331
F968  EDFF      01332         DA   HEAD
F96A            01333         DB   'FIND'
F96E  00        01334         NOP
F96F  CDD2F0    01335 FIND:   CALL GFITY
F972  CD00F1    01336         CALL ASHEX
F975  E5        01337         PUSH HL
F976  CD00F1    01338         CALL ASHEX
F979  CD55F1    01339         CALL FNEXT
```

```
F97C  ED5340FF  01340         LD   (TXTPT),DE       ;BEGINN DER ZU SUCH. BYTES
F980  E5        01341         PUSH HL               ;ENDADRESSE RETTEN
F981  3A08FF    01342         LD   A,(FITY)         ;OPTION.TESTEN UND BEFEHLSFOLGE ERZEUGEN
F984  21BFF9    01343         LD   HL,FINCH         ;ZEICHENKETTE
F987  1110FF    01344         LD   DE,DATA
F98A  010C00    01345         LD   BC,12
F98D  FE41      01346         CMP  'A'
F98F  2803      01347         JRZ  FIND1-#
F991  21C9F9    01348         LD   HL,FINHX         ;HEX
F994  EDB0      01349 FIND1:  LDIR                  ;BEFEHLSFOLGE INS RAM LADEN
F996  C1        01350         POP  BC               ;ENDADRESSE
F997  03        01351         INC  BC
F998  E1        01352         POP  HL               ;STARUTADRESSE
F999  ED5B40FF  01353 FIND2:  LD   DE,(TXTPT)
F99D  E5        01354         PUSH HL
F99E  C310FF    01355 FIND3:  JMP  DATA             ;VARIABLES PGM AUSFUEHREN
F9A1  BE        01356 FIND4:  CMP  (HL)
F9A2  23        01357         INC  HL
F9A3  28F9      01358         JRZ  FIND3-#          ;GLEICH,WEITER
F9A5  79        01359 FIND7:  LD   A,C              ;TEST OB ENDE
F9A6  95        01360         SUB  L
F9A7  78        01361         LD   A,B
F9A8  9C        01362         SBC  H
F9A9  E1        01363         POP  HL
F9AA  3010      01364         JRNC FIND6-#
F9AC  C9        01365         RET
F9AD  1A        01366 FIND8:  LD   A,(DE)
F9AE  FE0D      01367         CMP  CR
F9B0  20F3      01368         JRNZ FIND7-#
F9B2  E1        01369 FIND5:  POP  HL
```

```
F9B3  CD4AF1   01370          CALL OUTHL                 ;ADRESSE AUSGEBEN
F9B6  CD74F1   01371          CALL PRINT
F9B9           01372          DB   '  '
F9BB  00       01373          DB   0
F9BC  23       01374  FIND6:  INC  HL
F9BD  18DA     01375          JR   FIND2-#               ;WEITERSUCHEN
               01376
F9BF  1A       01377  FINCH:  LD   A,(DE)                ;STRING SUCHEN
F9C0  13       01378          INC  DE
F9C1  FE0D     01379          CMP  CR
F9C3  C2A1F9   01380          JPNZ FIND4                 ;VERGLEICHEN
F9C6  C3B2F9   01381          JMP  FIND5                 ;GEFUNDEN
F9C9  E5       01382  FINHX:  PUSH HL                    ;HEX SUCHEN
F9CA  CD00F1   01383          CALL ASHEX
F9CD  7D       01384          LD   A,L
F9CE  E1       01385          POP  HL
F9CF  D2A1F9   01386          JPNC FIND4                 ;VERGLEICHEN
F9D2  C3ADF9   01387          JMP  FIND8                 ;GEFUNDEN ??
               01388
               01389  ;******************************
               01390  ;* EOF                        *
               01391  ;******************************
               01392
F9D5  EDFF     01393          DA   HEAD
F9D7           01394          DB   'EOF'
F9DA  00       01395          NOP
F9DB  3AB3FF   01396  EOFI:   LD   A,(EOFCH)
F9DE  CD31F1   01397          CALL OUTH
F9E1  CDF8F0   01398          CALL INHEX
F9E4  D8       01399          RC
```

```
F9E5  7D        01400         LD   A,L
F9E6  32B3FF    01401         LD   (EOFCH),A
F9E9  C9        01402         RET
                01403
                01404  ;******************************
                01405  ;* POWER OFF                  *
                01406  ;******************************
                01407
F9EA  EDFF      01408         DA   HEAD
F9EC            01409         DB   'OFF'
F9EF  00        01410         NOP
F9F0  D300      01411  OFF:   OUT  0
F9F2  D300      01412         OUT  0
F9F4  C9        01413         RET
                01414
                01415  ;******************************
                01416  ;* ASNRI                      *
                01417  ;******************************
                01418
                01419
F9F5  EDFF      01420         DA   HEAD
F9F7            01421         DB   'ASNRI=RI'
F9FF  00        01422         NOP
FA00  218CFA    01423  RIRI:  LD   HL,RI1
FA03  180F      01424         JR   ASNRI-#
                01425
FA05  EDFF      01426         DA   HEAD
FA07            01427         DB   'ASNRI=CRI'
FA10  00        01428         NOP
FA11  21A3FA    01429  RICRI: LD   HL,CRI1
```

```
FA14  22C6FF   01430  ASNRI: LD   (RI+1),HL
FA17  C9       01431         RET
               01432
FA18  EDFF     01433         DA   HEAD
FA1A           01434         DB   'ASNRI=USR'
FA23  00       01435         NOP
FA24  CD00F1   01436         CALL ASHEX
FA27  D8       01437         RC
FA28  18EA     01438         JR   ASNRI-#
               01439
FA2A  3AB2FF   01440  MEMSI: LD   A,(MSIZE+1)          ;WIRD VON EXTERNEN PGM GENUTZT
FA2D  47       01441         LD   B,A
FA2E  3AB1FF   01442         LD   A,(MSIZE)
FA31  C9       01443         RET
               01444
               01445  ;EINFACHER RAMTEST
FA32  21FFFF   01446  MEMCH: LD   HL,-1                ;START BEI 0
FA35  1100E8   01447         LD   DE,SYSBG             ;BIS SYSTEMBEGINN
FA38  23       01448  MCH:   INC  HL
FA39  7E       01449         LD   A,(HL)
FA3A  EEFF     01450         XOR  OFFH                 ;KOMPLEMENTIEREN
FA3C  77       01451         LD   (HL),A
FA3D  BE       01452         CMP  (HL)
FA3E  200C     01453         JRNZ MCH1-#
FA40  EEFF     01454         XOR  OFFH
FA42  77       01455         LD   (HL),A
FA43  BE       01456         CMP  (HL)                 ;WIEDER GLEICH?
FA44  2006     01457         JRNZ MCH1-#               ;NEIN
FA46  E5       01458         PUSH HL
FA47  ED52     01459         SBC  HL,DE                ;ENDE?
```

```
FA49  E1        01460          POP  HL
FA4A  20EC      01461          JRNZ MCH-#               ;NEIN,WEITER
FA4C  2B        01462   MCH1:  DEC  HL
FA4D  C9        01463          RET
                01464
                01465   ;ROUTINE FUER EXTERNE MONITORBENUTZUNG
FA4E  E5        01466   EXT:   PUSH HL
FA4F  C5        01467          PUSH BC
FA50  F5        01468          PUSH AF
FA51  21A9FB    01469          LD   HL,ETAB
FA54  0600      01470          LD   B,0
FA56  CB21      01471          SLA  C                   ;C=C*2
FA58  09        01472          ADD  HL,BC
FA59  7E        01473          LD   A,(HL)
FA5A  23        01474          INC  HL
FA5B  66        01475          LD   H,(HL)
FA5C  6F        01476          LD   L,A                 ;HL=ZIELADRESSE
FA5D  F1        01477          POP  AF
FA5E  C1        01478          POP  BC
FA5F  E3        01479          EX   (SP),HL             ;ZIEL==>STACK,HL ZURUECK
FA60  C9        01480          RET                      ;JMP
                01481
                01482
                01483   ;****************************
                01484   ;* GERAETEROUTINEN          *
                01485   ;****************************
                01486
                01487
                01488   ;KASSETTE INITIALISIEREN
FA61  0E19      01489   CINIT: LD   C,CSIOA
```

```
FA63  0608     01490         LD   B,8
FA65  2173FA   01491         LD   HL,CTAB
FA68  EDB3     01492         OTIR
FA6A  3E05     01493         LD   A,5
FA6C  D316     01494         OUT  CTC2
FA6E  3E04     01495         LD   A,4
FA70  D316     01496         OUT  CTC2
FA72  C9       01497         RET
               01498
FA73  04       01499  CTAB:  DB   4
FA74  04       01500         DB   4
FA75  01       01501         DB   1
FA76  40       01502         DB   40H
FA77  03       01503         DB   3
FA78  C1       01504         DB   0C1H
FA79  05       01505         DB   5
FA7A  EA       01506         DB   0EAH
               01507
               01508  ;LOCHBANDLESEN INITIALISIEREN
FA7B  3E4F     01509  RINIT: LD   A,04FH
FA7D  D30D     01510         OUT  CP1A
FA7F  3ECF     01511         LD   A,0CFH
FA81  D30F     01512         OUT  CP1B
FA83  3E01     01513         LD   A,1
FA85  D30F     01514         OUT  CP1B
FA87  3E10     01515         LD   A,10H
FA89  D30F     01516         OUT  CP1B
FA8B  C9       01517         RET
               01518
               01519  ;ZEICHEN VON LOCHBAND HOLEN
```

```
FA8C  3E10     01520 RI1:   LD   A,10H                 ;STARTIMPULS
FA8E  D30E     01521        OUT  DP1B
FA90  AF       01522        XOR  A
FA91  D30E     01523        OUT  DP1B
FA93  CDB8FA   01524 RI12:  CALL SEOF
FA96  DB0E     01525        IN   DP1B
FA98  E601     01526        AND  1
FA9A  20F7     01527        JRNZ RI12-#
FA9C  DB0C     01528        IN   DP1A
FA9E  E67F     01529        AND  7FH
FAA0  28EA     01530        JRZ  RI1-#
FAA2  C9       01531        RET
               01532
               01533 ;ZEICHEN VON KASSETTE HOLEN
FAA3  CDB8FA   01534 CRI1:  CALL SEOF
FAA6  DB19     01535        IN   CSIOA
FAA8  E601     01536        AND  1
FAAA  28F7     01537        JRZ  CRI1-#
FAAC  DB18     01538        IN   DSIOA
FAAE  C9       01539        RET
               01540 ;ZEICHEN AUF KASSETTE AUFZEICHNEN
FAAF  D318     01541 CP01:  OUT  DSIOA
FAB1  DB19     01542 CP011: IN   CSIOA
FAB3  E604     01543        AND  4
FAB5  28FA     01544        JRZ  CP011-#
FAB7  C9       01545        RET
               01546
FAB8  DB08     01547 SEOF:  IN   KEYBD                 ;LESEN UNTERBRECHEN?
FABA  FE03     01548        CMP  EOF
FABC  C0       01549        RNZ
```

```
FABD  F1      01550         POP  AF              ;STACK KORRIGIEREN
FABE  3AB3FF  01551         LD   A,(EOFCH)
FAC1  C9      01552         RET
              01553
              01554  ;CONSOLSTATUSABFRAGE
FAC2  DB09    01555  CSTS:  IN   KEYBC
FAC4  B7      01556         OR   A
FAC5  C8      01557         RZ                   ;KEINE TASTE
FAC6  DB08    01558         IN   KEYBD
FAC8  FE0D    01559         CMP  CR
FACA  28F6    01560         JRZ  CSTS-#
FACC  32B4FF  01561         LD   (CSTSR),A
FACF  F6FF    01562         OR   0FFH
FAD1  C9      01563         RET
              01564
              01565  ;TASTATUREINGABE
FAD2  3AB4FF  01566  CI1:   LD   A,(CSTSR)
FAD5  B7      01567         OR   A
FAD6  2807    01568         JRZ  CI11-#
FAD8  AF      01569         XOR  A
FAD9  32B4FF  01570         LD   (CSTSR),A
FADC  DB08    01571         IN   KEYBD
FADE  C9      01572         RET
FADF  DB09    01573  CI11:  IN   KEYBC           ;AUF STEIGENDE FLANKE WARTEN
FAE1  B7      01574         OR   A
FAE2  20FB    01575         JRNZ CI11-#
FAE4  DB09    01576  CI12:  IN   KEYBC
FAE6  B7      01577         OR   A
FAE7  28FB    01578         JRZ  CI12-#
FAE9  DB08    01579         IN   KEYBD
```

```
FAEB  C9        01580         RET
                01581
                01582  ;BILDSCHIRMAUSGABE
FAEC  E5        01583  C01:   PUSH HL
FAED  D5        01584         PUSH DE
FAEE  C5        01585         PUSH BC
FAEF  F5        01586         PUSH AF
FAF0  2A93FF    01587         LD   HL,(CURS)         ;HL=CURSOR
FAF3  47        01588         LD   B,A
FAF4  3E80      01589         LD   A,80H             ;CUSOR AUS
FAF6  AE        01590         XOR  (HL)
FAF7  77        01591         LD   (HL),A
FAF8  78        01592         LD   A,B
FAF9  FE1F      01593         CMP  NL+1              ;STEUERZEICHEN?
FAFB  302B      01594         JRNC CRT5-#            ;NEIN
FAFD  FE1E      01595         CMP  NL                ;NEUE ZEILE?
FAFF  114000    01596         LD   DE,CHPL           ;ZEICHEN/ZEILE
FB02  2007      01597         JRNZ CRT2-#            ;NEIN,NICHT NL
FB04  7D        01598         LD   A,L               ;ZUM ZEILENANFANG ZURUECK
FB05  E6C0      01599         AND  -CHPL
FB07  6F        01600         LD   L,A
FB08  19        01601  CRT1:  ADD  HL,DE             ;LINEFEED
FB09  181F      01602         JR   CRT7-#
FB0B  FE0D      01603  CRT2:  CMP  CR
FB0D  2006      01604         JRNZ CRT3-#
FB0F  7D        01605         LD   A,L               ;ZUM ZEILENANF. ZURUECK
FB10  E6C0      01606         AND  -CHPL
FB12  6F        01607         LD   L,A
FB13  1838      01608         JR   CRT9-#
FB15  FE0A      01609  CRT3:  CMP  LF
```

```
FB17  28EF     01610        JRZ   CRT1-#            ;LINEFEED AUSFUEHREN
FB19  FE08     01611        CMP   CURL
FB1B  380B     01612        JRC   CRT5-#
FB1D  203A     01613        JRNZ  CRT10-#
FB1F  2B       01614        DEC   HL                ;CURSOR LINKS
FB20  7C       01615 CRT4:  LD    A,H               ;TEST OB NOCH IM BILDSCHIRM
FB21  FEE8     01616        CMP   H(CRTBG)
FB23  DA84FB   01617        JPC   CRT15
FB26  1825     01618        JR    CRT9-#
FB28  77       01619 CRT5:  LD    (HL),A            ;ZEICHEN NACH CRT
FB29  23       01620 CRT6:  INC   HL                ;CURSOR RECHTS
FB2A  7C       01621 CRT7:  LD    A,H               ;ROLLEN NOTWENDIG?
FB2B  FEEE     01622        CMP   H(CRTEN)
FB2D  381E     01623        JRC   CRT9-#            ;NEIN
FB2F  1100E8   01624        LD    DE,CRTBG
FB32  E5       01625        PUSH  HL
FB33  2140E8   01626        LD    HL,CRTBG+CHPL
FB36  010006   01627        LD    BC,CRTEN-CRTBG
FB39  EDB0     01628        LDIR                    ;ROLLEN
FB3B  21C0ED   01629        LD    HL,CRTEN-CHPL
FB3E  0640     01630        LD    B,CHPL
FB40  3E20     01631        LD    A,' '
FB42  77       01632 CRT8:  LD    (HL),A            ;LETZTE ZEILE LOESCHEN
FB43  23       01633        INC   HL
FB44  10FC     01634        DJNZ  CRT8-#
FB46  E1       01635        POP   HL
FB47  114000   01636        LD    DE,CHPL           ;CURSOR KORRIGIEREN
FB4A  B7       01637        OR    A
FB4B  ED52     01638        SBC   HL,DE
FB4D  3E80     01639 CRT9:  LD    A,80H             ;CURSOR AN
```

```
FB4F  AE       01640         XOR  (HL)
FB50  77       01641         LD   (HL),A
FB51  2293FF   01642         LD   (CURS),HL
FB54  F1       01643         POP  AF
FB55  C1       01644         POP  BC
FB56  D1       01645         POP  DE
FB57  E1       01646         POP  HL
FB58  C9       01647         RET
FB59  FE09     01648  CRT10: CMP  CURR
FB5B  28CC     01649         JRZ  CRT6-#
FB5D  FE0B     01650         CMP  CURU
FB5F  2005     01651         JRNZ CRT11-#
FB61  B7       01652         OR   A                  ;CURSOR HOCH
FB62  ED52     01653         SBC  HL,DE
FB64  18BA     01654         JR   CRT4-#
FB66  FE1B     01655  CRT11: CMP  ESC
FB68  2005     01656         JRNZ CRT12-#
FB6A  7E       01657         LD   A,(HL)             ;INVERTIEREN
FB6B  EE80     01658         XOR  80H
FB6D  18B9     01659         JR   CRT5-#
FB6F  FE0C     01660  CRT12: CMP  CLS
FB71  200D     01661         JRNZ CRT14-#
FB73  2100E8   01662  CRT19: LD   HL,CRTBG
FB76  3620     01663  CRT13: LD   (HL),' '           ;LOESCHEN
FB78  23       01664         INC  HL
FB79  7C       01665         LD   A,H
FB7A  FEEE     01666         CMP  H(CRTEN)
FB7C  20F8     01667         JRNZ CRT13-#            ;ANSCHLIESSEND HOME
FB7E  3E0E     01668         LD   A,HOME
FB80  FE0E     01669  CRT14: CMP  HOME
```

```
FB82  20A4     01670          JRNZ CRT5-#
FB84  2100E8   01671  CRT15:  LD   HL,CRTBG
FB87  18C4     01672          JR   CRT9-#
               01673
               01674  ;TABELLE FUER GERAETEZUWWEISUNG
FB89  FFEF     01675  ATAB:   DA   START-1            ;MSIZE
FB8B  04       01676          DB   EOT                ;EOFCH
FB8C  00       01677          DB   0                  ;CSTSR
FB8D  79       01678          LD   A,C
FB8E  C3ECFA   01679          JMP  C01                ;L0
FB91  79       01680          LD   A,C
FB92  C3AFFA   01681          JMP  CP01               ;P0
FB95  79       01682          LD   A,C
FB96  C3AFFA   01683          JMP  CP01               ;CP0
FB99  79       01684          LD   A,C
FB9A  C3ECFA   01685          JMP  C01                ;C0
FB9D  C38CFA   01686          JMP  RI1
FBA0  C3A3FA   01687          JMP  CRI1
FBA3  C3D2FA   01688          JMP  CI1
FBA6  00F0     01689          DA   START
FBA8  C3       01690          DB   0C3H
               01691
FBA9  F8F0     01692  ETAB:   DA   INHEX              ;0
FBAB  00F1     01693          DA   ASHEX              ;1
FBAD  20F1     01694          DA   CNVBN              ;2
FBAF  31F1     01695          DA   OUTH               ;3
FBB1  4AF1     01696          DA   OUTHL              ;4
FBB3  74F1     01697          DA   PRINT              ;5
FBB5  81F1     01698          DA   INL                ;6
FBB7  87F1     01699          DA   INL00              ;7
```

```
FBB9  CFF1    01700        DA    MEMO1          ;8
FBBB  40F3    01701        DA    MOVE           ;9
FBBD  63F8    01702        DA    READ1-3        ;10 =0AH
FBBF  6EF3    01703        DA    LDEXT          ;11 =0BH
FBC1  DAF4    01704        DA    SAEXT          ;12 =0CH
FBC3  F0F9    01705        DA    OFF            ;13 =0DH
FBC5  00FA    01706        DA    RIRI           ;14 =0EH
FBC7  11FA    01707        DA    RICRI          ;15 =0FH
FBC9  61FA    01708        DA    CINIT          ;16 =10H
FBCB  7BFA    01709        DA    RINIT          ;17 =11H
FBCD  DDF5    01710        DA    FILL1          ;18 =12H
FBCF  25F6    01711        DA    NMIV           ;19 =13H
FBD1  55F6    01712        DA    UPUSH          ;20 =14H
FBD3  73F6    01713        DA    UPOP           ;21 =15H
FBD5  19F7    01714        DA    REGDP          ;22 =16H
FBD7  9FF7    01715        DA    REG1           ;23 =17H
FBD9  02F9    01716        DA    CHECK          ;24 =18H
FBDB  6FF9    01717        DA    FIND           ;25 =19H
FBDD  32FA    01718        DA    MEMCH          ;26 =1AH
              01719
              01720
              01721
              01722        EJEC
              01723  ;******************************
              01724  ;* VEREINBARUNGEN             *
              01725  ;******************************
              01726
              01727  ;PORTS:
              01728  POWOF: EQU  0                  ;POWER OFF
              01729  KEYBD: EQU  8                  ;TASTATURDATEN
```

```
01730  KEYBC:  EQU  9       ;TASTATURSTATUS 0FFH=GEDR./0=KEINE TASTE
01731  BEPOF:  EQU  8       ;SIGNALTON AUS
01732  BEPON:  EQU  9       ;AN
01733  DP1A:   EQU  0CH     ;12  PIO 1
01734  CP1A:   EQU  0DH     ;13
01735  DP1B:   EQU  0EH     ;14
01736  CP1B:   EQU  0FH     ;15
01737  DP2A:   EQU  10H     ;16  STEUERPORT LBL
01738  CP2A:   EQU  11H     ;17  BIT 0 STATUS,BIT 4 STARTIMPULS
01739  DP2B:   EQU  12H     ;18 LBL DATEN
01740  CP2B:   EQU  13H     ;19
01741  CTC0:   EQU  14H     ;20
01742  CTC1:   EQU  15H     ;21
01743  CTC2:   EQU  16H     ;22 KASSETTE
01744  CTC3:   EQU  17H     ;23
01745  DSI0A:  EQU  18H     ;24 KASSETTE
01746  CSI0A:  EQU  19H     ;25
01747  DSI0B:  EQU  1AH     ;26
01748  CSI0B:  EQU  1BH     ;27
01749  ;
01750  CURL:   EQU  8       ;CURSOR LINKS
01751  CURR:   EQU  9       ;RECHTS
01752  CURD:   EQU  0AH     ;10 ;RUNTER
01753  CURU:   EQU  0BH     ;11 ;HOCH
01754  HOME:   EQU  0EH     ;14
01755  ESC:    EQU  1BH     ;ESCAPE
01756  NL:     EQU  1EH     ;NEW LINE
01757  LF:     EQU  0AH     ;LINE FEED
01758  CR:     EQU  0DH     ;RETURN
01759  CLS:    EQU  0CH     ;BILDSCHIRM LOESCHEN
```

```
          01760  SOH:   EQU  1           ;START KOPF
          01761  GS;    EQU  1DH         ;GRUPPENTRENNUNG
          01762  ETX:   EQU  03H         ;END OF TEXT
          01763  EOT:   EQU  04H         ;END OF TRANSMISSION
          01764  STX:   EQU  02H         ;START OF TEXT
          01765  ERASE: EQU  06H         ;EINGABEZEILE LOESCHEN
          01766  EOF:   EQU  ETX         ;END OF FILE
          01767  MEMB:  EQU  16          ;BYTES/ZEILE (MEMORY)
          01768  MEML:  EQU  16          ;ANZAHL ZEILEN (MEM)
          01769  CHPL:  EQU  64          ;ZEICHEN/ZEILE (CRT)
          01770  HEAD:  EQU  0FFEDH      ;SUCHCODE VOR KOMMANDO
          01771
          01772         ORG  0FF00H
          01773  ;RAM BEREICH
          01774  STACK: EQU  #-2
          01775  USTCK: EQU  #-64
FF00      01776  FINA:  BER  8           ;FILENAME REGISTER
FF08      01777  FITY:  BER  2           ;FILETYP REGISTER
FF0A      01778  ARG1:  BER  2           ;HEX EINGABEN
FF0C      01779  ARG2:  BER  2
FF0E      01780  ARG3:  BER  2
FF10      01781  DATA:  BER  16          ;ZUSATZREGISTER (FILE-INFO
FF20      01782  INTV:  BER  32          ;INTERRUPT VEKTOREN
FF40      01783  TXTPT: BER  2           ;POINTER IN TEXTPUFFER
FF42      01784  TXTBU: BER  81          ;TEXTPUFFER
          01785  BULEN: EQU  81          ;PUFFERLAENGE
FF93      01786  CURS:  BER  2           ;CURSORPOSITION
FF95      01787  HAF:   BER  2           ;CPU-REGISTER
FF97      01788  HBC:   BER  2
FF99      01789  HDE:   BER  2
```

```
FF9B         01790  HHL:    BER  2
FF9D         01791  AAF:    BER  2
FF9F         01792  ABC:    BER  2
FFA1         01793  ADE:    BER  2
FFA3         01794  AHL:    BER  2
FFA5         01795  HIX:    BER  2
FFA7         01796  HIY:    BER  2
FFA9         01797  HIF:    BER  2
             01798  UREG:   EQU  HAF
FFAB         01799  USP:    BER  2
FFAD         01800  UPC:    BER  2
FFAF         01801  SSP:    BER  2                ;ZWISCHENSPEICHER FUER SP
FFB1         01802  MSIZE:  BER  2                ;HOECHSTER FREIER SPEICHERPLATZ
FFB3         01803  EOFCH:  BER  1                ;EOF ZEICHEN
FFB4         01804  CSTSR:  BER  1                ;CONSOLSTATUSZELLE
FFB5         01805  LOE:    BER  1                ;LD A,C
FFB6         01806  LO:     BER  3                ;JMP LIST-DEVICE
FFB9         01807  POE:    BER  1                ;LD A,C
FFBA         01808  PO:     BER  3                ;JMP PUNCH-DEVICE
FFBD         01809  CPOE:   BER  1                ;LD A,C
FFBE         01810  CPO:    BER  3                ;JMP KASSETTENAUSGABE
FFC1         01811  COE:    BER  1                ;LD A,C
FFC2         01812  CO:     BER  3                ;JMP CONSOLAUSGABE
FFC5         01813  RI:     BER  3                ;JMP LBL
FFC8         01814  CRI:    BER  3                ;JMP KASSETTENEINGABE
FFCB         01815  CI:     BER  3                ;JMP CONSOLEINGABE
FFCE         01816  BRKPT:  BER  3                ;BREAKPOINTMERKER
             01817
             01818  CRTBG:  EQU  0E800H           ;BILDSCHIRMADR.
             01819  CRTEN:  EQU  0EE00H           ;ENDE+1
```

```
                   01820  MEMMO: EQU  DATA             ;DARSTELLMODE BEI MEMORY
                   01821  SYSBG: EQU  CRTBG
                   01822
                   01823
  FFD1             01824         END

NEXT: E

PASS: E
```

Anmerkungen zum Programm

Die in Spalte Code nicht ausgedruckten Byte können der folgenden Aufstellung entnommen werden.

```
FO38 20 48 2E 4D 4F 4E 20 56 32 2E 31 20 31 32 2F 38 33
FO5C 49 4C 4C 45 47 41 4C 20 43 4F 4D 4D 41 4E 44
F1C2 4D 45 4D 4F 52 59
F200 20 20 43 55 52 53 4F 52 3A
F237 20 20
F309 52 41 4D 20 45 52 52 4F 52 20
F31A 4D 4F 56 45
F369 4C 4F 41 44
F3CD 46 49 4C 45 20 46 4F 55 4E 44 20
F490 20 42 41 44 20 46 49 4C 45 20
F4A4 20 45 52 52 4F 52 53
F4D5 53 41 56 45
F599 49 4E
F5A7 4F 55 54
F5B7 46 49 4C 4C
F5EE 53 49 5A 45
F5F6 53 49 5A 45 3A
F605 20 52 41 4D 54 4F 50 3A
F621 4E 4D 49
F642 42 52 45 41 4B 20 41 54 20
F696 47 4F
F71C 20 41 46 20 20 20 42 43 20 20 20 44 45 20 20 20 48
F72D 4C 20 20 20 41 46
F734 20 20 42 43
F739 20 20 44 45
E73E 20 20 48 4C
F743 20 20 49 58 20 20 20 49 59 20 20 49 20 20 20 20 53
F754 50 20 20 20 20 50 43
F780 20 20
F79B 52 45 47
F809 46 41 43 42 45 44 4C 48
F813 48 45 4C 50
F826 20 20 20
F852 52 45 41 44
F8AE 43 48 45 43 4B 53 55 4D 20 45 52 52 4F 52 20
F8E9 43 48 45 43 4B 53 55 4D
F91B 3D
```

F92C 53 55 4D 3A
F93B 20 20 44 49 46 3A
F94F 20 20 52 45 4C 3A
F964 2A 2A
F96A 46 49 4E 44
F9B9 20 20
F9D7 45 4F 46
F9EC 4F 46 46
F9F7 41 53 4E 52 49 3D 52 49
FA07 41 53 4E 52 49 3D 43 52 49
FA1A 41 53 4E 52 49 3D 55 53 52

10. Hardwareaufbau und Inbetriebnahme

10.1. Stromversorgung

Bild 10.1 zeigt die Schaltung für ein Netzteil, das alle für den Mikrocomputer benötigten Betriebsspannungen liefert.
Zusätzlich wird die Gleichspannung -12 V bereitgestellt, die für den Betrieb analoger Schaltungselemente (vor allem Operationsverstärker) erforderlich ist. Die Einzelschaltungen wurden im wesentlichen [12] entnommen und erfüllen hinsichtlich Strombelastbarkeit und Konstanz der Ausgangsspannungen alle Anforderungen der vorliegenden Anwendung.
Wie Bild 10.1 zeigt, wurden die Regelstrecken für die Ausgangsgleichspannungen auf der Basis integrierter Spannungsregler aufgebaut. Die Betriebsspannung wird damit mit minimalem Aufwand erzeugt. Für das Regelteil der bezüglich des Ausgangsstroms am meisten belasteten Betriebsspannung + 5 V wurde der Spannungsreglerschaltkreis *MAA 723* eingesetzt. Die weiteren Betriebsspannungen erzeugen die integrierten Festspannungsregler *MA 7805* bzw. *MA 7812*. Auf Grund der verwendeten Spannungsregler (bei + 5 V in Verbindung mit geeigneter externer Beschaltung) sind alle Regelstrecken kurzschlußfest und temperaturkompensiert.
Ohne Schutzmaßnahmen können die Schaltkreise *MA 7805* bzw. *MA 7812* bei entsprechender kapazitiver Belastung zerstört werden, denn beim Abschalten entladen sich Lastkapazitäten über die genannten Regelschaltkreise. Zum Schutz der Schaltkreise wurden deshalb die Dioden VD9, VD14, VD19 vorgesehen.
Der Einfluß netzbedingter Störungen läßt sich durch Netzfilter wirkungsvoll vermindern.
Im Monitorprogramm ist das softwaremäßige Abschalten des Computers über die Tastatur vorgesehen (Monitorbefehl: OFF). Wer von dieser Möglichkeit Gebrauch machen will, muß natürlich einen kleinen zusätzlichen Aufwand im Netzteil in Kauf nehmen. Bild 10.2 zeigt die für diesen Zweck benötigte Relaisschaltung.
Im Zusammenhang mit der Stromversorgung ist es notwendig,

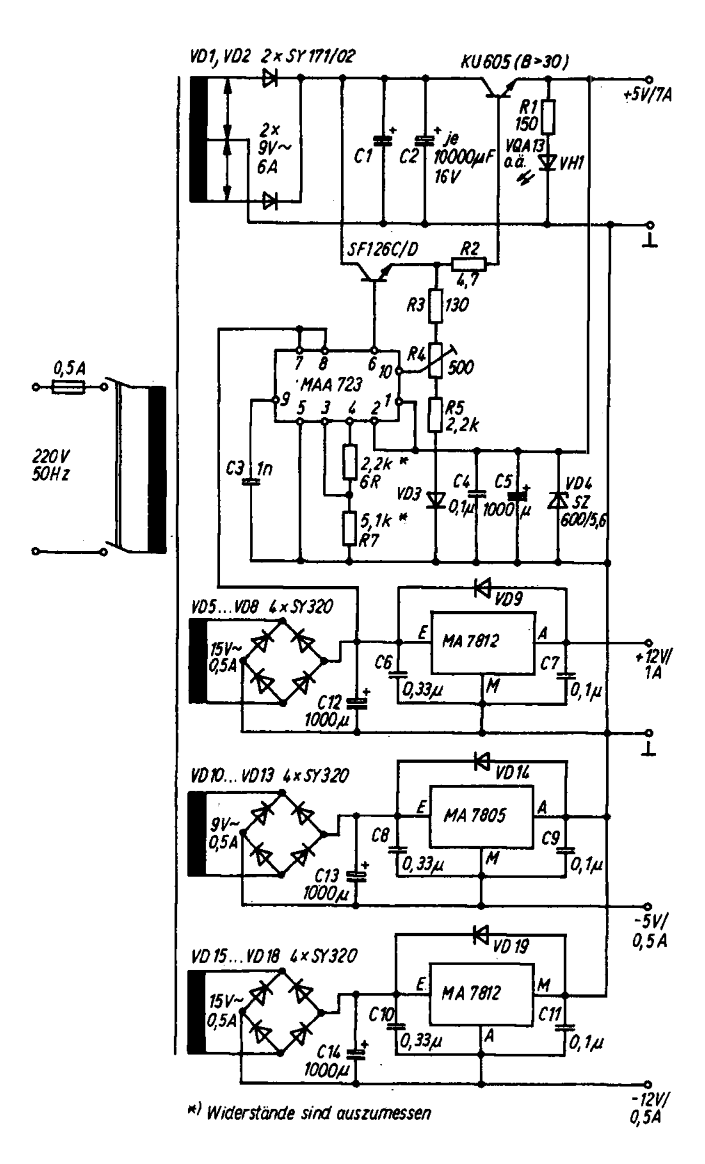


Bild 10.1 Netzteil für 5 V, – 5V, 12 V, –12 V

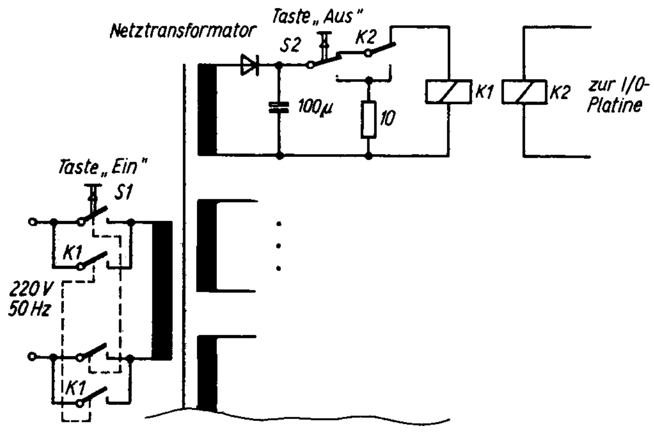


Bild 10.2 Relaisschaltung für Power-off

noch auf die bereits im Abschnitt 4.3.5 erwähnte wichtige Randbedingung für den Einsatz der Speicherschaltkreise *U 555 (2708)* bzw. *U 256 (4116)* hinzuweisen:
Eine Vorrangschaltung muß absichern, daß die Betriebsspannung $U_{BB} = -5$ V (Substratvorspannung für oben genannte Speichertypen) beim Einschalten zuerst und beim Ausschalten zuletzt am Schaltkreis anliegt. Beim Ausfall von U_{BB} muß man die Betriebsspannungen $U_{CC} = 5$ V und $U_{DD} = 12$ V sofort abschalten. Beachtet man diese Vorschrift nicht, können die genannten Schaltkreise zerstört werden. Nach den Erfahrungen der Autoren ist die Beachtung der genannten Randbedingung dringend anzuraten.
Bild 10.3 zeigt eine für diesen Zweck bewährte Vorrangschaltung. Beim praktischen Aufbau dieser Schaltung ist darauf zu achten, daß die Spannung − 5 V unmittelbar am Bus abgegriffen wird (d. h. in der Nähe der gefährdeten Speicherschaltkreise).

10.2. Systemaufbau

Sämtliche Daten-, Adreß- und Steuerleitungen einschließlich Taktsignal C müssen auf den Systembus geführt werden. Das gilt auch für die Betriebsspannungen (+ 5 V, − 5 V, + 12 V, − 12 V) und die Masseleitung.

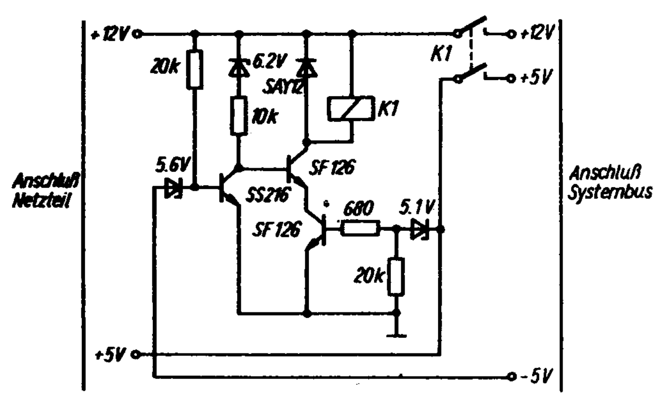


Bild 10.3 Vorrangschaltung für −5 V

Die konkrete Gestaltung des Systembusses ist abhängig von den Gegebenheiten, die dem einzelnen zur Verfügung stehen. Das bezieht sich besonders auf Steckverbinder, Einschub- und damit Leiterkartenabmessungen sowie Möglichkeiten zum Anfertigen der Leiterkarten. Unabhängig von diesen individuellen Voraussetzungen sind noch einige allgemeingültige Hinweise notwendig.
So muß vor allem auf Systematik und Übersichtlichkeit geachtet werden. Von der Funktion her zusammengehörige Signale sollten auch auf dem Bus benachbart und in geordneter Reihenfolge untergebracht werden, sofern übergeordnete Gesichtspunkte dem nicht entgegenstehen. Weiter achtet man möglichst darauf, daß Kurzschlüsse benachbarter Leitungen nicht zu Zerstörungen führen. Betriebsspannungen und Masse werden zweckmäßigerweise doppelt kontaktiert.
Besonders wichtig ist es, daß man die Störungsfreiheit auf dem Systembus absichert. Störungen sind sowohl unerwünschte Beeinflussungen von Bussignalen untereinander als auch von außen kommende Einwirkungen (z. B. vom Netz oder durch Störstrahlung verursacht). Unter dem zuletzt genannten Aspekt sind alle Busleitungen so kurz wie möglich zu halten. Netzspannungen führende Leitungen dürfen grundsätzlich nicht in Busnähe angeordnet werden.
Eine Orientierung für die Systembusgestaltung bieten industrielle Busausführungen, z. B. die des *K 1520* [13]. Aus der Sicht des

Amateurs sind jedoch individuelle von Industriestandards abweichende Busgestaltungen zulässig. Wenn dabei die oben genannten Gesichtspunkte beachtet werden, ist bei der Systemtaktfrequenz von 2,5 MHz ein problemloser Betrieb im allgemeinen abgesichert.
Für die praktische Realisierung erweist es sich am günstigsten, den Bus in gedruckter Schaltungstechnik als Rückverdrahtungsplatine aufzubauen. Dabei werden die Steckverbinder direkt auf die Busplatine gelötet. Betriebsspannungs- und Masseleitungen versieht man entsprechend ihrer Strombelastung mit größerem Querschnitt.
Als Ergänzung zu dieser Problematik sei noch darauf hingewiesen, daß gemäß den Erfahrungen der Autoren ein konventionell verdrahteter Systembus in sauberer Ausführung, unter Beachtung oben genannter Gestaltungsgesichtspunkte, ebenfalls einwandfrei funktioniert.
Der vorgestellte Mikrocomputer wurde in Einschubbauweise realisiert. Im Computereinschub werden die CPU-Platine sowie Speicher- und Interfacebaugruppen untergebracht und über den Systembus miteinander verbunden. Eine Ausnahme macht das Tastaturinterface, das (abgesehen vom Tastatur-PROM) direkt in der Tastatur plaziert wird. Das ist deshalb sinnvoll, weil andernfalls für das Verbindungskabel zwischen Tastatur und Computer wesentlich mehr Leitungen benötigt werden (etwa 32 an Stelle von 11).
Der Computereinschub enthält auch die Stromversorgung. Reservesteckplätze ermöglichen eine spätere individuelle Erweiterung.
Abmessung und Gestaltung der Leiterkarten für die im Computereinschub befindlichen Baugruppen sind, wie bereits erwähnt, von den individuellen Möglichkeiten des einzelnen abhängig. Aus diesem Grund werden in dieser Broschüre auch keine Leiterplattenentwürfe veröffentlicht.
Als günstiges Leiterkartenformat haben sich die EGS-Abmessungen 170 mm × 210 mm erwiesen. Frei verdrahtbare Universalplatinen sollte man nicht verwenden. Der Aufwand, der mit dem Entwurf und der Herstellung sauberer, übersichtlicher Leiterkarten verbunden ist, zahlt sich spätestens bei Inbetriebnahme und Fehlersuche sowie durch hohe Betriebszuverlässigkeit wieder aus. Es sei davor gewarnt, die doch vergleichsweise komplizierte Hardware eines Mikrocomputers auf der Basis von „Drahtverhau“-

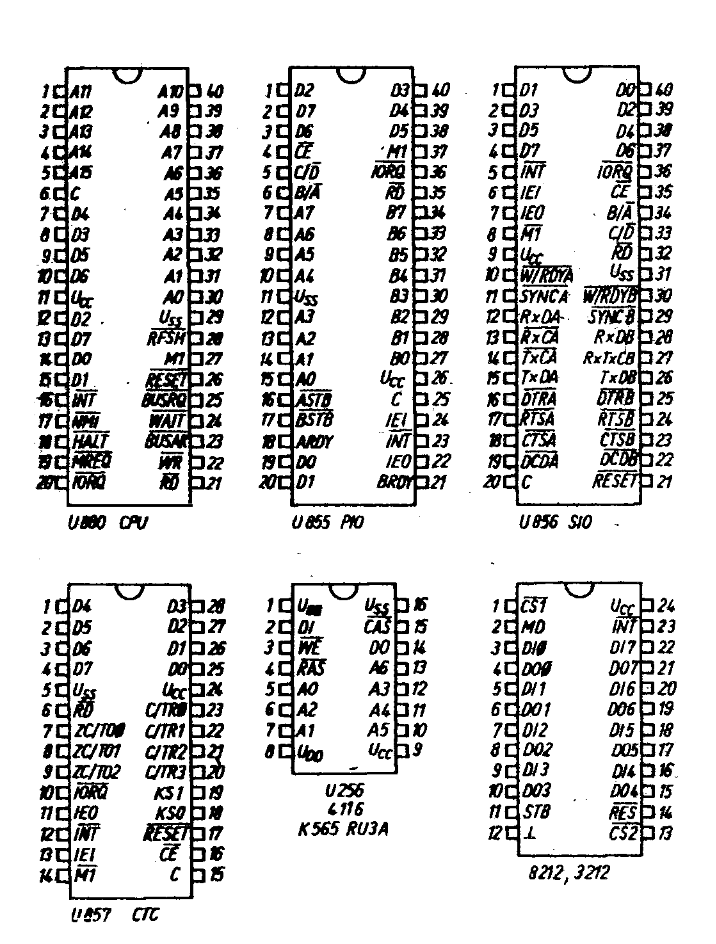
Schaltungen zu realisieren. Bei einem derartigen Schaltungsaufbau wird außer dem permanenten Mißerfolg am Computer nichts programmierbar sein.
Auf Grund der computerspezifischen Schaltungen stellt man die Platinen nur aus doppelt kaschiertem Leiterplattenmaterial her. Zusammen mit den Leiterkartenabmessungen ergibt sich damit die Möglichkeit, alle wichtigen Baugruppen jeweils auf einer Leiterkarte unterzubringen.
Als Verbindungselemente zwischen Bus und Leiterkarten werden geeignete Steckverbinder benötigt. Zu empfehlen sind 58polige Steckverbinder indirekter Ausführung. Bei Verwendung direkter Steckverbinder sollten die Leiterplattenkontakte möglichst oberflächenveredelt sein. Dies ist dem Amateur aber nicht immer möglich.
Zum Aufbau des Mikrocomputers werden relativ viele Schaltkreise benötigt. Der Einsatz von qualitätsgeminderten Bastlerschaltkreisen kann zu erheblichen Problemen führen und bedeutet letztlich keine Kostenersparnis. Eine CPU *U880* mit verminderter Taktfrequenz (z. B. $f = 1$ MHz) kann man einsetzen, wenn der Taktgenerator dann entsprechend dimensioniert wird.
Für das Abblocken der Betriebsspannungszuführungen mit Stützkondensatoren gelten, wenn keine speziellen Hinweise gegeben wurden (siehe Abschnitt 4.2.2), die für TTL-Schaltkreise gültigen Richtlinien [3].

Wichtiger Hinweis zum Bildschirm:
Grundsätzlich darf man nur Fernsehgeräte verwenden, die galvanisch vom Netz getrennt sind. Das trifft im allgemeinen für moderne voll halbleiterbestückte Geräte zu. Empfohlen seien vor allem Koffergeräte. Besonders geeignet ist z. B. der Kofferempfänger *Combivision* wegen seiner guten Wiedergabequalität.
Unter Berücksichtigung aller Hinweise sollte der Mikrocomputer nun schrittweise in folgender Reihenfolge aufgebaut und in Betrieb gesetzt werden:
1. Einschub mit Systembus und zugehörigen Steckverbindern; für jede einzelne Leitung müssen der galvanische Durchgang über den gesamten Bus bis hin zum einzelnen Steckkontakt sowie die Kurzschlußfreiheit überprüft werden
2. Netzteil unter Beachtung der einschlägigen Schutzgütebestimmungen

3. CPU-Baugruppe
4. Fernsehinterface
5. Tastatur
6. Programmieren des Monitors auf PROM-Schaltkreise *U 555*, Inbetriebnahme des Monitorprogramms
7. Ein-/Ausgabeschnittstellen
8. Kassetteninterface
9. Speichererweiterung nach individuellen Möglichkeiten und Bedarf (möglichst viel RAM)
10. Weitere Hardware nach Bedarf (z. B. PROM-Programmiergerät)

Die in 10. angeführte Hardware sowie darüber hinausgehende individuelle Erweiterungen sollten erst in Angriff genommen werden, wenn die Punkte 1. bis 9. des Systemaufbaus zufriedenstellend absolviert sind. Voraussetzung für die Verwirklichung eigener Ideen ist natürlich auch die vorherige Aneignung ausreichender Hardware- und Softwarekenntnisse.
Zum Schluß sei noch der Hinweis gegeben, daß zur erfolgreichen Arbeit mit dem Mikrocomputer eine übersichtliche und ständig aktualisierte Hard- und Softwaredokumentation gehört. Erfahrungsgemäß wird nämlich genau das mühsam erarbeitete und eingetippte Programm versehentlich gelöscht, das man vergessen hat, ordentlich zu dokumentieren.

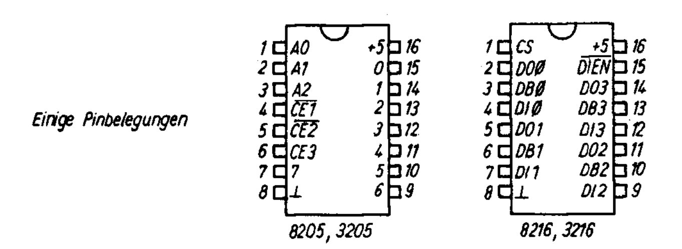
11. Anhang



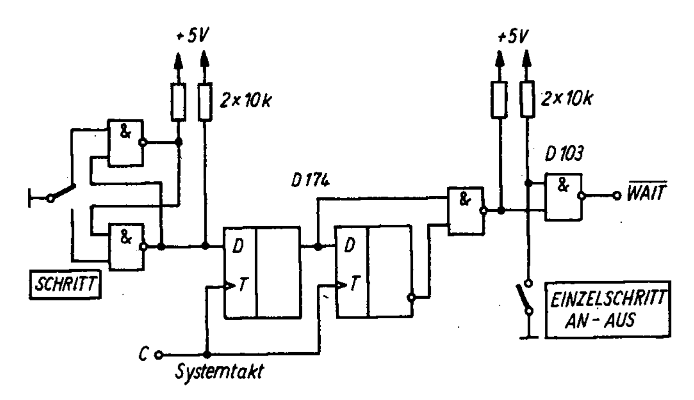
U880 CPU
A11 A12 A13 A14 A15 C D4 D3 D5 D6 Ucc D2 D7 D0 D1 INT NMI HALT MREQ IORQ
A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 Uss RFSH M1 RESET BUSRQ WAIT BUSAK WR RD
U855 PIO
D2 D7 D6 CE C/D B/A A7 A6 A5 A4 Uss A3 A2 A1 A0 ASTB BSTB ARDY D0 D1
D3 D4 D5 M1 IORQ RD B7 B6 B5 B4 B3 B2 B1 B0 Ucc C IEI INT IEO BRDY
U856 SIO
D1 D3 D5 D7 INT IEI IEO M1 Ucc W/RDYA SYNCA RxDA RxCA TxCA TxDA DTRA RTSA CTSA DCDA C
D0 D2 D4 D6 IORQ CE B/A C/D RD Uss W/RDYB SYNCB RxDB RxTxCB TxDB DTRB RTSB CTSB DCDB RESET
U857 CTC
D4 D5 D6 D7 Uss RD ZC/TO0 ZC/TO1 ZC/TO2 IORQ IEO INT IEI M1
D3 D2 D1 D0 Ucc C/TR0 C/TR1 C/TR2 C/TR3 KS1 KS0 RESET CE C
U256
4116
K565 RU3A
Ubb DI WE RAS A0 A2 A1 Udd
Uss CAS DO A6 A3 A4 A5 Ucc
8212, 3212
CS1 MD DI0 DO0 DI1 DO1 DI2 DO2 DI3 DO3 STB ⊥
Ucc INT DI7 DO7 DI6 DO6 DI5 DO5 DI4 DO4 RES CS2

EPROM 16k×8 27128	EPROM 8k×8 2764	EPROM 4k×8 2732	RAM 2k×8 6116	EPROM 2k×8 2716	RAM 1k×8 4118	EPROM 1k×8 U555 2708	Pin (28)	Pin (24)	Pin (24)	Pin (28)	EPROM 1k×8 U555 2708	RAM 1k×8 4118	EPROM 2k×8 2716	RAM 2k×8 6116	EPROM 4k×8 2732	EPROM 8k×8 2764	EPROM 16k×8 27128
Upp	Upp						1			28						+5	+5
A12	A12						2			27						$\overline{PGM}$	$\overline{PGM}$
A7	A7	A7	A7	A7	A7	A7	3	1	24	26	+5	+5	+5	+5	+5	NC	A13
A6	A6	A6	A6	A6	A6	A6	4	2	23	25	A8	A8	A8	A8	A8	A8	A8
A5	A5	A5	A5	A5	A5	A5	5	3	22	24	A9	A9	A9	A9	A9	A9	A9
A4	A4	A4	A4	A4	A4	A4	6	4	21	23	-5	$\overline{WE}$	Upp	$\overline{WE}$	A11	A11	A11
A3	A3	A3	A3	A3	A3	A3	7	5	20	22	$\overline{CS}$/WE	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$/Upp	$\overline{OE}$	$\overline{OE}$
A2	A2	A2	A2	A2	A2	A2	8	6	19	21	+12	$\overline{L}$	A10	A10	A10	A10	A10
A1	A1	A1	A1	A1	A1	A1	9	7	18	20	Prog	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
A0	A0	A0	A0	A0	A0	A0	10	8	17	19	D7	D7	D7	D7	D7	D7	D7
D0	D0	D0	D0	D0	D0	D0	11	9	16	18	D6	D6	D6	D6	D6	D6	D6
D1	D1	D1	D1	D1	D1	D1	12	10	15	17	D5	D5	D5	D5	D5	D5	D5
D2	D2	D2	D2	D2	D2	D2	13	11	14	16	D4	D4	D4	D4	D4	D4	D4
⊥	⊥	⊥	⊥	⊥	⊥	⊥	14	12	13	15	D3	D3	D3	D3	D3	D3	D3

Pinbelegung einiger Byte-Wide-Speicher



Einige Pinbelegungen



Einfache Einzelschrittschaltung, mit dessen Hilfe Hardwarefehler aufgespürt werden können. Die Signale auf dem Datenbus und auf dem Adressbus können durch Leuchtdioden oder Hexadezimalanzeigen sichtbar gemacht werden.

HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII
00	NUL	10	DLE	20	SPACE	30	0	40	@	50	P	60	\	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$ ¤	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	–	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	∧↑	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_←	6F	o	7F	DEL

Umwandlungstabelle Hexadezimal ⟷ ASCII-Code

12. Literaturverzeichnis

[1] *Barthold, H., Bäurich, H.*. Mikroprozessoren – Mikroelektronische Schaltkreise und ihre Anwendung (Teil 1, 2, 3); H. 202, 203 und 204 der Reihe „electronica", Berlin 1982

[2] *Kieser, H., Meder, M.:* Mikroprozessortechnik, Berlin 1982

[3] *Kühn, E., Schmied, H.:* Handbuch Integrierter Schaltkreise; Berlin 1978

[4] *Entress, G., Knobloch, G., Meusel, K.-H.:* Fernsehkompatibles Datensichtgerät als Ausgabeeinrichtung für Klein- und Mikrorechner. In: radio fernsehen elektronik 27 (1978) H. 12, S. 796 – 800

[5] *Hertzsch, A.:* Einfacher Impulsbreitendiskriminator. In: radio fernsehen elektronik 32 (1983) H. 2, S. 121

[6] Datenblatt *U 555*

[7] Datenblatt Firma Intel

[8] *Licht, U.:* PROMs mit geringem Schaltungsaufwand programmiert. In: Elektronik 27 (1978) H. 8, S. 61 – 63

[9] ...: Basis-MOS für Mikrorechner *K 1520*; Geräteroutine für *K 0420* (Stand 28. 02. 1981). VEB Kombinat Robotron

[10] *Fliegenbaum, W.:* D-A-Umsetzer mit geringem Linearitätsfehler. In: radio fernsehen elektronik 31 (1982) H. 5

[11] *Döring, H.:* Analogwerteingabe in Mikrorechner mit *C 520 D.* In: radio fernsehen elektronik 31 (1982) H. 6, S. 382 – 383; Berichtigung In: radio fernsehen elektronik 31 (1982) H. 9, S. 549

[12] *Jungnickel, H.:* Stromversorgungseinrichtungen; Berlin 1982

[13] ...: Betriebsdokumentation Mikrorechnersystem *K 1520.* VEB Kombinat Robotron